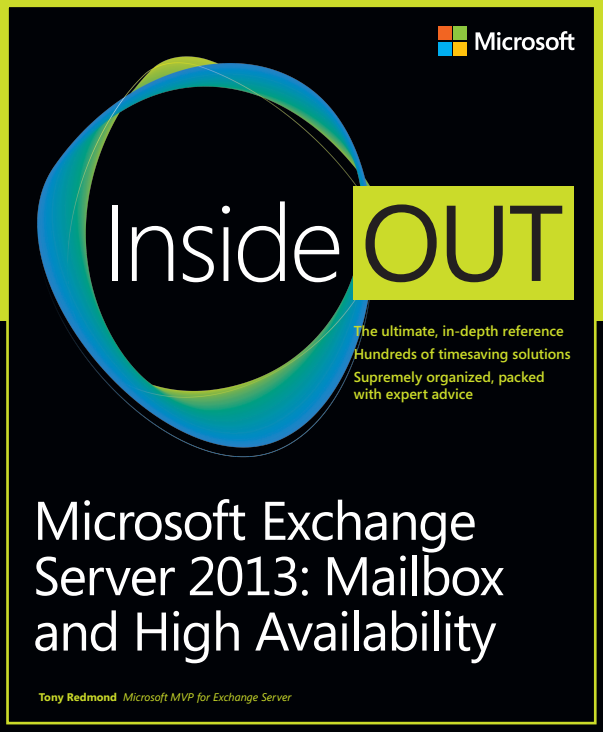
EXCERPT



Chapter 11
Compliance

Tony Redmond

Microsoft

Compliance: EXCERPT from Microsoft® Exchange Server 2013 Inside Out

Tony Redmond

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2013948704
ISBN: 978-0-7356-8086-9

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: nSight, Inc.
Technical Reviewer: Paul Robichaux; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Kerin Forsyth
Indexer: Lucie Haskins
Cover: Twist Creative • Seattle

Contents at a Glance

Introduction

When Paul Robichaux and I first sat down to discuss how we might cooperate in writing Microsoft Exchange Server 2013 Inside Out, we were conscious that the sheer complexity and breadth of the product meant that many months of fact-gathering, writing, checking, and editing would be necessary to produce a book that described Exchange 2013 in sufficient depth and detail to warrant the "Inside Out" title. In fact, we knew that two books were necessary to avoid ending up with one 1,600-page mega-volume, so we divided the task to align with the Mailbox and Client Access Server roles, which is the plan that we've used to create the books.

At the same time, we knew that there were particular parts of Exchange 2013 that justified their own book. I've often felt that Unified Messaging was a very misunderstood part of the product. Paul has worked in this area for many years and has taught the subject to students aspiring to become Microsoft Certified Masters for Exchange. He's the best possible person to write about Unified Messaging. I have had a particular interest in compliance, and it's an area in which Microsoft has invested massively over Exchange 2010 and Exchange 2013, so it was easy (relatively) to create a mini-book on this topic. Then we were faced with High Availability, an area that is so important to so many companies who depend on their email being available all the time. The advent of the Database Availability Group (DAG) was a tremendously important advance for the product in Exchange 2010, and it's even better in Exchange 2013. Making High Availability the focus of the third mini-book made a lot of sense.

Each mini-book is a chapter from the larger "Inside Out" title, but each stands on its own merits and can be read in isolation. However, if you really want to get acquainted with Exchange 2013, you might just want to check out the two-volume set. We think you'll like it.

Errata & book support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site:

http://aka.ms/ExIOv1/errata

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://www.microsoft.com/learning/booksurvey

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

Paul is available on Twitter *@PaulRobichaux*. His blog is available at *http://paulrobichaux.wordpress.com/*. Tony is available on Twitter *@12Knocksinna*, while his blog is at *http://thoughtsofanidlemind.wordpress.com/*.

CHAPTER 11

Compliance management

The need to comply with legal and regulatory requirements is a fact of corporate life today. Legislation such as the Sarbanes–Oxley Act in the United States has influenced many other countries to introduce similar requirements to keep records that show when something was done and by whom. Microsoft started on the process to build messaging records management (MRM) capability into Exchange Server 2007. However, users (and many administrators) didn't understand the purpose of MRM, and take-up was weak. To address these issues and to provide a true basis for compliance, Exchange 2010 introduced a wide range of new features, including a new implementation of MRM and the first appearance of archive mailboxes. Exchange 2013 builds on the foundation laid by Exchange 2010 to refine and improve the features, including the following:

- The ability to audit administrator and mailbox actions.
- The provision of archive mailboxes as a replacement for the much-loathed (at least by me) PSTs.
- The ability to create and apply retention policies to items and folders.
- The ability to recover items without requiring a backup to be restored.
- The ability to conduct searches against mailbox content, including deleted items information held in the Recoverable Items folder. If Exchange 2013 is integrated with Microsoft SharePoint 2013, searches can also be conducted against information held on SharePoint sites through the eDiscovery Center. If Exchange 2013 operates on a standalone basis, eDiscovery searches are conducted through the Exchange Administration Center (EAC).
- The ability to place mailboxes on hold so that items are retained even when the user might prefer them not to be.

The features just listed are not exhaustive because other Exchange features can be associated with compliance. For example, transport rules allow a disclaimer to be appended to every outgoing message that can limit liability by complying with rules that say messages from a company must contain specific contact or other information about the company. Data loss prevention (DLP) is a new feature of Exchange 2013, which uses a special form of transport rules and some special processing to detect patterns in email content that might represent different forms of confidential information, to stop users from sending this information through email when they should not. The transport system also supports journal rules and enables servers to capture copies of messages in a legally robust manner so that those copies can be cited in court cases. (Capturing information through an immutable hold is another way to achieve the same goal.) Transport, journal, and DLP rules are covered in *Exchange Server 2013 Inside Out: Connectivity, Clients, and UM*, by Paul Robichaux (Microsoft Press, 2013).

When you examine the range of compliance features incorporated in Exchange, you understand that compliance is an area that continues to evolve, largely because changing legal and regulatory needs drive such evolution. Microsoft will update the Exchange feature set over time to satisfy the broadest possible set of requirements. However, it is unreasonable to expect any software to deliver a complete answer. For example, although you can place mailboxes on hold or establish an extended deleted item retention period to ensure that important information is not deleted, you might also need to develop administrative procedures to handle situations such as mailbox retention following the death or dismissal of an employee. Handling situations such as this could be reasonably straightforward—disable the account, hide the user's mailbox from the Global Address List (GAL), and keep it and any archives, including PSTs that you recover from the user's PC, until any legal hold period has passed—as long as you think everything through. With that thought in mind, let's review the compliance features in Exchange 2013.

The joy of legal discovery

Legal discovery actions have been around for centuries. Over the past two decades, the focus of discovery or searches for information pertinent to a legal case has begun to shift from paper evidence to electronic evidence. This shift reflects the different manner in which organizations store data today. Filing cabinets are still stuffed with paper, but much of the correspondence companies conducted by letter, fax, and telex are now sent by email, so the focus of discovery has to accommodate both paper and electronic media.

Discovery actions for email systems first began in the mid-1980s. Messages were recovered from backup tapes and printed for lawyers to review. The process was expensive and time consuming. The only mitigating factor was that it was much easier to determine who might have sent an incriminating message because relatively few people in a company had email, and the overall volume of email was low. Messages were text only and tended to be short.

It was therefore possible to satisfy a judge's order to retrieve all messages for 10 specific users over a month without running up an extraordinarily high bill.

Today's environment is different. Many more users are typically hosted on each server, they send and receive an ever-increasing volume of messages, and those messages contain many types of attachments, including video and audio files. The result of living in the age of electronic communication is that the cost of legal discovery is higher because there is more information to process. In March 2009, *Fortune* magazine reported that the court-appointed trustee of bankrupt Lehman Brothers Inc. had captured 3.2 billion email and instant messages, occupying 1.4 terabytes (TB). This isn't an unusual amount; the FBI investigation of Enron in 2001 reviewed 31 TB of data and used 4 TB as evidence. Email is a critical means of business communication that has replaced telexes, faxes, and written letters in many respects, so legal discovery of email has moved from an out-of-the-ordinary situation to a form that is extremely common, whether it is to satisfy a legal or regulatory requirement, respond to a subpoena, or deal with an internal matter concerning employee ethics, harassment, or discipline.

The first generation of Exchange offered no way to store mail after it was deleted, so you had to restore a database from a backup if you wanted to recover a message, whether it was needed to satisfy a legal order or because a user had deleted it in error. Gradually, Microsoft began to add new features to Exchange to help. The original version of the dumpster (the official term now used is the "Recoverable Items" structure), as implemented in Exchange 2000 through Exchange 2007, provides a two-phase delete process by which messages are marked as deleted but kept in the database until their retention period expires, at which time they are removed. As discussed in the "The function of the Recoverable Items structure" section later in this chapter, Exchange 2013 uses an enhanced set of folders to underpin a number of compliance features such as in-place hold.

Journaling appeared in Exchange 2003 and was upgraded in Exchange 2007. However, the journaling functionality Exchange offered was basic, and most companies that invested in products to capture copies of messages preferred purpose-designed products such as Symantec's Enterprise Vault or Iron Mountain's NearPoint. As mentioned earlier, Microsoft added managed folders in Exchange 2007 with the idea that administrators could create folders that are distributed to mailboxes for users to store important items. However, the reality is that most organizations ignored managed folders.

The compliance features in Exchange 2007 were a start. However, the overall experience was not compelling enough to generate widespread usage, which then led Microsoft to create a new set of features that have been rolled out over Exchange 2010 and Exchange 2013.

Although Exchange 2013 includes a wide range of compliance features, Microsoft must convince customers that having integrated archiving and search incorporated in an email

server is a better solution than dedicated archiving and search applications that have been in use and developed over many years. It can be argued that cost is one key Microsoft advantage because archiving is available at the price of an enterprise Client Access License (CAL) that might be already acquired. Another obvious advantage is the integration of the compliance features into the core of Exchange, meaning that customers do not have to pay for and manage an additional system to gain compliance features.

The cost of an enterprise CAL for each user is often lower than the cost of dedicated archiving software plus any additional hardware that is required to run the archiving software. This argument works only if the functionality available in Exchange meets your requirements. Microsoft acknowledges that many vendors have been actively selling compliance solutions for Exchange for nearly a decade. Some offer different functionality than Exchange, especially in areas such as workflow, the ability to archive information taken from other sources, and the experience that companies have with these products in integrating compliance processes with various regulations.

If SharePoint and Exchange are the most important repositories of information within your company, the two will serve as an excellent platform to enable compliance, provided that you can deploy the necessary software versions to use features such as site mailboxes and conduct searches across both repositories. (See *http://technet.microsoft.com/en-us/library/jj218665.aspx* for information about how to configure Exchange for use with the SharePoint eDiscovery Center.) If Exchange 2013 is used without SharePoint, then the focus needs to be on how to extract maximum advantage from its compliance features. Based on this premise, you then focus on:

- Deploying archive mailboxes in an attempt, perhaps in vain, to eliminate the sprawl of PSTs used across the company. The aim is to use large mailboxes to enable users to keep all their data online, which is an advantage for both users and the company. After it is online, the data is exposed to indexing and search.

- Deploying suitable retention policies to help users keep control of their (now larger) mailboxes. Retention policies can sweep unwanted items out of user mailboxes on a regular and automatic basis while moving items that need to be retained into archive mailboxes.

- Working with the company legal department to determine appropriate policies to govern:

    - When users are placed on hold (when they are prevented from deleting items from their mailboxes or making any other alteration to mailbox content). Exchange captures attempts to delete or edit information in the backup without interfering with the user's ability to work with her mailbox.

- When and how eDiscovery searches are performed, who can authorize these operations, who has access to the data recovered by searching, how long this data is retained, and how and when it is removed from servers.
- When administrator and mailbox auditing is used and who has access to reports generated from this data.

Having some focused goals for compliance is a good way to begin complying. With that point in mind, the following discusses some of the ways you can comply.

Archive mailboxes

An archive mailbox, or personal archive, is a logical extension of a user's primary mailbox that provides an online archive facility. The name might cause some confusion with the personal archives users create with Microsoft Outlook. The big difference is that the Exchange archive is tightly integrated in the Information Store, and the data held in the archive are therefore accessible using all the features available to mailboxes, including eDiscovery searches. By comparison, PST archives are usually confined to an individual PC, and the data that they contain are inaccessible to server-based processing. (It is possible, but not recommended and not supported, to place PSTs on network file shares; see *http://support.microsoft.com/kb/297019/en-us*.)

An archive mailbox can be stored in the same database as the primary mailbox, or it can be in a different database. Some deployments have created special archive servers that host databases containing only archive mailboxes. This is a perfectly acceptable solution that offers some advantages because the hardware can be tailored to the lower demands that exist for access to archive information. Usually, people don't access their archive mailboxes as frequently as they do a primary mailbox, which is constantly busy with the process of receiving and sending messages. In essence, therefore, an archive is infrequently used but always available online.

INSIDE OUT **Outlook cached Exchange mode and archive mailboxes**

Outlook maintains local copies of all the folders in the primary mailbox through synchronization when it operates in cached Exchange mode to enable the user to work offline or to work unhindered by any temporary network outage. Outlook never caches local copies of folders from archive mailboxes. An archive is designed to hold information that is important and has to be retained but is not accessed frequently. Therefore, you can access information in an archive mailbox only when you are connected online to Exchange.

If you use Microsoft Office 365, archive mailboxes can be stored in the cloud, an option that has proven increasingly attractive as companies gain more experience and confidence with cloud-based services. It is attractive to hive off archives to a cloud-based service because this enables you to remain focused on the care and maintenance of production mailboxes while the hosting provider takes care of the archives. Whatever option you choose, a mailbox can have just one personal archive, and each mailbox that has an on-premises archive requires an enterprise CAL. Mailboxes that use cloud-based archive mailboxes in the Microsoft Exchange Online Archive service do not need enterprise CALs.

Microsoft views archive mailboxes as the natural replacement for PSTs. The growth of messages and the reluctance of administrators to increase mailbox quotas coupled with the inability of Exchange and its clients to deal elegantly with very large mailboxes (5 GB and up) meant that most organizations were forced to use PSTs to offload data from the online store. Users do like to keep messages, even if they never look at them again. (Some conference speakers have opined that a message filed in a PST has a 99 percent chance of never being looked at again after six months; my personal experience tallies with this estimate.) Other problems with PST management typically cited in corporate messaging deployments include the following:

- **Reduced security** PSTs are personal stores, but users keep just about anything in them, including sensitive and usually unencrypted corporate information ranging from budgets to presentations about new products to performance reviews. If someone loses a laptop—or even a USB device that has a PST on it—that information is immediately exposed and potentially available to anyone who finds the device and accesses it. Even if protected by a password, the PST file structure is insecure and can be quickly accessed by using utilities commonly available on the Internet. After the password is bypassed, a PST can be opened using any Microsoft Outlook client.

- **Inability to respond to discovery actions** Information held on a PST is usually invisible to searches that a company performs to respond to discovery requests. This is fine if the information is personal or irrelevant to the discovery request, but it could be very expensive if required information is not disclosed to a court and is subsequently discovered.

- **Inability to apply policy** Many companies have a data retention policy that requires users to delete documents and messages after a certain period. The period can vary, depending on the type of information contained in different items. In any case, the company loses any ability to apply policy centrally after a user moves an item from his mailbox into a PST.

- **Exposure to data loss** Laptop disks are notoriously prone to failure. If users don't back up their data, any disk crash exposes them to potential data loss, and that information might be important.

The alternative solution to increasing disk quota for mailboxes in previous versions of Exchange was to buy and deploy a dedicated third-party archiving solution. Using PSTs is obviously far cheaper for a company. It's also easier for users because they control how many PSTs they create and how they use them. Some create a separate PST for each year; some create a PST for each major project. The big downside is that PSTs then expose the company to the risks previously described. Even so, it will take time to pry user fingers from their beloved PSTs.

Exchange archive mailboxes are not perfect, and a number of limitations exist that could hinder deployment, including the following:

- You cannot transfer an archive to another mailbox. If a user leaves and you delete her mailbox, the archive is also removed. You can save data by exporting items from the archive (and the primary mailbox) to a PST and then importing it back into the personal archive of another user, but it would be more elegant just to transfer the archive intact.

- You cannot copy or move sections of the archive to transfer it to another user. For example, a user who wants to transfer responsibility for a project to another user has to extract the folders and other items relating to the project from her archive and provide them to the other user. Again, the workaround is to export selected folders from the personal archive to a PST and provide the PST to the other user (or import the PST into her archive). Alternatively, a site mailbox or public folder might serve as a better repository for information that has to be shared between different project members.

- You cannot assign permissions on a folder level within the archive to allow users to give access to parts of their archive to other users. Delegates who have full access to a user's mailbox can access the complete archive for that mailbox.

- Archive mailboxes are inaccessible from mobile clients and from Outlook for Mac. Given the use of mobile devices today, this can be an issue for some users.

These are examples of functionality Microsoft will doubtless consider enhancing in the future. It's likely Microsoft will wait to see how archives are used in practical terms within customer deployments before it plans how archives evolve in future releases of Exchange.

Enabling archives

Before you can create and use archive mailboxes with Exchange, you have to deploy clients that support the feature. These are as follows:

- Outlook 2007 Service Pack 3 (running the November 2012 cumulative update or later; see *http://blogs.technet.com/b/exchange/archive/2010/12/20/3411710.aspx*)

- Outlook 2010 Service Pack 1 (running the November 2012 update or later)
- Outlook 2013
- Outlook Web App

Outlook Professional Plus and Outlook standalone support archive mailboxes (see *http://office.microsoft.com/en-us/outlook-help/license-requirements-for-personal-archive-and-retention-policies-HA102576659.aspx)*. The other editions (such as Office Home and Business 2013) do not include the code necessary to open and reveal archive mailboxes. No mobile clients currently support access to archive mailboxes.

You can enable an archive when you create a new mailbox by clicking the More Options link at the bottom of the screen used to enter new mailbox details. This reveals the check box by which to indicate that an archive should be created alongside the primary mailbox (Figure 11-1). You can also select a specific mailbox database to hold the archive or just click Save to have Exchange use its auto-provisioning feature to select a database to hold the archive.

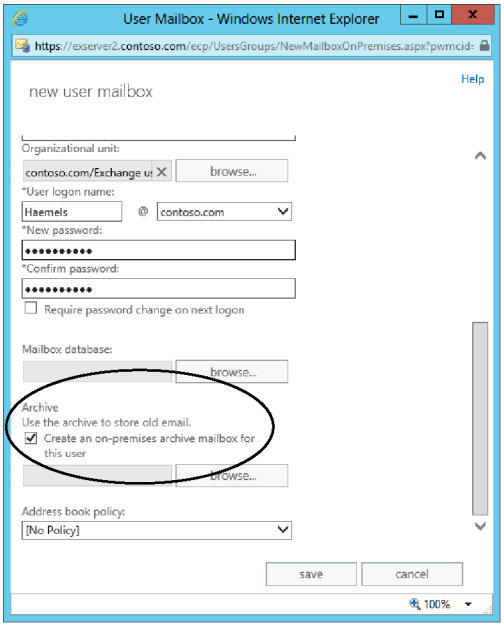


Figure 11-1 Opting to create an archive for a new mailbox

To enable an archive when you create a mailbox with Exchange Management Shell (EMS), you just add the –Archive parameter to the New-Mailbox cmdlet and the –ArchiveDatabase

parameter to select a database for the archive mailbox. (Again, this is not necessary because Exchange will pick a database for you if you omit the –ArchiveDatabase parameter.)

See Chapter 5, "Mailbox management," for a full discussion about how to create new mailboxes.

INSIDE OUT You won't lose access to your archive if there is a server failure

EAC restricts the databases you can choose to hold an archive for a mailbox to those that are mounted on servers in the same site. However, Exchange supports an exception to this rule when a database transfers to a server in another site following a failure. In this case, the CAS redirects clients to the archive in the database on the other site by using a cross-site connection for as long as the database is hosted on that site. You won't want to use cross-site connections for an extended period, and normal connections will resume after you switch the database that contains the personal archive back to a server on its original site. If you have a tenant subscription for Office 365 and have configured the necessary hybrid environment to support data moving between on-premises and cloud servers, you will also see the option displayed by EAC to allow Office 365 to host an archive.

You can also enable an archive for existing mailboxes by selecting a mailbox in EAC and then selecting Enable under the In-Place Archive section in the action pane (Figure 11-2). EAC displays a dialog box to enable you to select a database for the archive and to warn you that enabling this feature requires an enterprise CAL. If you click OK, the mailbox is enabled with an archive. You can also enable a personal archive for an existing mailbox with EMS. For example:

```
Enable-Mailbox –Identity 'Tony Redmond' –Archive
```

As soon as an archive has been enabled for a mailbox, it becomes available to Outlook Web App and Outlook the next time the client refreshes its resource information through the Autodiscover process, which provides Outlook with information about the new archive. This usually takes a few minutes for on-premises mailboxes but might need up to an hour for an archive hosted in Office 365. Outlook Web App retrieves information about the archive when it connects to the mailbox online. At this point, the new archive will hold only a Deleted Items folder.

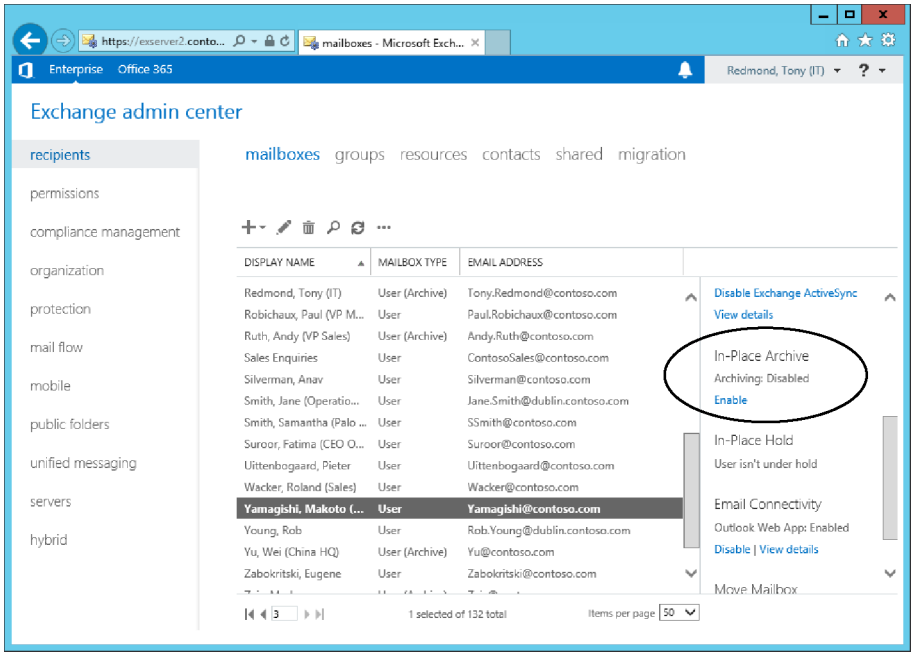


Figure 11-2 Enabling an archive for an existing mailbox

After you enable an archive for a mailbox, you'll notice that EAC displays User (Archive) instead of just User in the mailbox type column. Unlike its Exchange Management Console (EMC) predecessor, EAC does not include a prewritten query to display all the mailboxes quickly that currently have an archive. Because EAC regards archive-enabled mailboxes as having a different type, you can click the mailbox type heading to have EAC sort the mailboxes and present them together. Alternatively, you can use the Get-Recipient or Get-Mailbox cmdlets to search for mailboxes that have an archive. For example, this command looks for mailboxes that have archives enabled and reports the mailbox name, the archive name, and the databases in which the mailbox and archive are located:

```
Get-Mailbox –Archive | Format-Table Name, ArchiveName, Database, ArchiveDatabase
```

Mailboxes that use MRM 1.0 cannot have an archive

Although Microsoft does not recommend it, you can enable an archive for room, equipment, and shared mailboxes, which seems a little strange. However, you cannot enable an archive for a mailbox that has been assigned a managed folders policy. Managed folders provide the basis for messaging records management in Exchange 2007, but they are superseded by retention policies from Exchange 2010 onward and therefore do not support archive mailboxes.

Managing archive properties

Behind the scenes, EAC calls the Enable-Mailbox cmdlet to enable an archive. These commands first enable the personal archive for a mailbox and then retrieve the properties that Exchange maintains for an archive.

```
Enable-Mailbox –Identity 'Andy.Ruth@contoso.com' –Archive
Get-Mailbox –Identity 'Andy.Ruth@contoso.com' | Select Name, Arch*
```

```
ArchiveDatabase         : DB2
ArchiveGuid             : 03c8b429-5160-4418-868c-2816b8a31d71
ArchiveName             : {Personal Archive - Ruth, Andy (VP Sales)}
ArchiveQuota            : 50 GB (53,687,091,200 bytes)
ArchiveWarningQuota     : 45 GB (48,318,382,080 bytes)
ArchiveDomain           :
ArchiveStatus           : None
ArchiveState            : Local
ArchiveRelease          :
```

The first four archive properties are always populated for a mailbox when it is archive-enabled. The globally unique identifier (GUID) identifies the archive mailbox within the database where it is stored. The default name for the archive is derived from the Personal Archive prefix plus the mailbox's display name and can be changed afterward to whatever name you prefer. The archive quotas are inherited from the default values set for the database and reflect the values Exchange uses to limit the amount of information in the archive and when it starts to issue warning messages.

You can alter these values with the Set-Mailbox cmdlet. For example:

```
Set-Mailbox –Identity 'Andy.Ruth@contoso.com' –ArchiveName "Andy's Splendid Online
Archive" –ArchiveQuota 2GB –ArchiveWarningQuota 1.9GB –UseDatabaseQuotaDefaults
$False
```

The last four properties have the following meaning:

- *ArchiveDomain* is used only if the personal archive is stored on an Exchange Online server (Office 365). If used, the property holds the name of the tenant domain.
- *ArchiveStatus* contains a status value to indicate whether the personal archive has been created on an Exchange Online server.
- *ArchiveState* is Local, in this case meaning that the archive is on a local, on-premises server.
- *ArchiveRelease* is reserved for Microsoft purposes and might be used to indicate that an archive depends on a particular version of Exchange in the future. For now, it remains blank.

Checking space usage

The amount of space used in an archive mailbox can be checked with the Get-MailboxStatistics cmdlet, which supports the –Archive parameter to tell it to report details of the archive mailbox rather than the primary mailbox. For example:

```
Get-MailboxStatistics –Identity 'John Smith' –Archive | Format-Table DisplayName,
ItemCount, TotalItemSize, LastLogonTime -AutoSize
```

Updating properties of an archive mailbox

You can update some archive properties, including its name, through EAC. To do this, select the mailbox, click Edit, navigate to Mailbox Features, and then select View Details for the Archiving section. You can update the name (Figure 11-3) and the quotas assigned to the archive. EAC also displays details of the quota currently used in the archive.

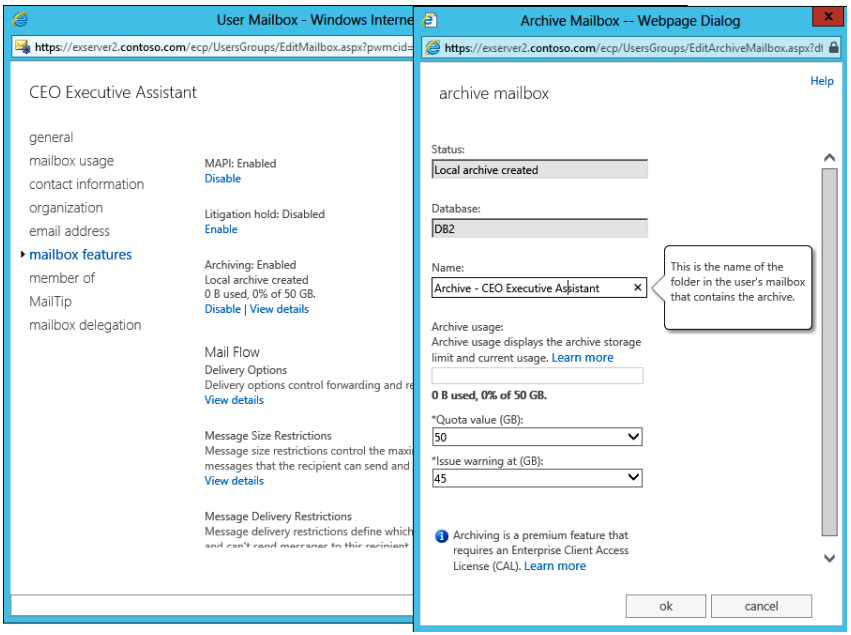


Figure 11-3 Updating the name of a personal archive through EAC

The default archive and retention policy

When you enable a personal archive for a mailbox, Exchange assigns a retention policy called Default MRM Policy to the mailbox unless the mailbox has a retention policy already assigned. This policy is designed to help the mailbox's owner use the archive by automatically moving items from the primary mailbox into the archive as their retention period

expires. Assigning a retention policy automatically to a mailbox might be considered helpful, but the action has some consequences for users, which are discussed in the following paragraphs.

The retention period applied by the default tag in the policy is two years, so the effect of applying the policy is that any item in the mailbox that is not stamped with another tag will be moved into the archive after it is two years old. The retention policy assigned to the mailbox becomes effective the next time the Managed Folder Assistant (MFA) processes the mailbox. The default policy is not assigned if the mailbox is already under the control of another retention policy. How to manipulate retention policies and tags is discussed in the "Messaging records management" section later in this chapter.

The name of the default retention policy Exchange supplies has changed over the versions; if you previously deployed Exchange 2010, you will find that another policy, the Default Archive and Retention Policy, has been applied to mailboxes that had archives enabled with Exchange 2010. In fact, an even earlier policy, Default Archive Policy, might also be present. This policy was provided with the original release to manufacturing (RTM) of Exchange 2010. Even if the other two policies are available, Exchange will apply the Exchange 2013 version to mailboxes that are newly enabled with archives.

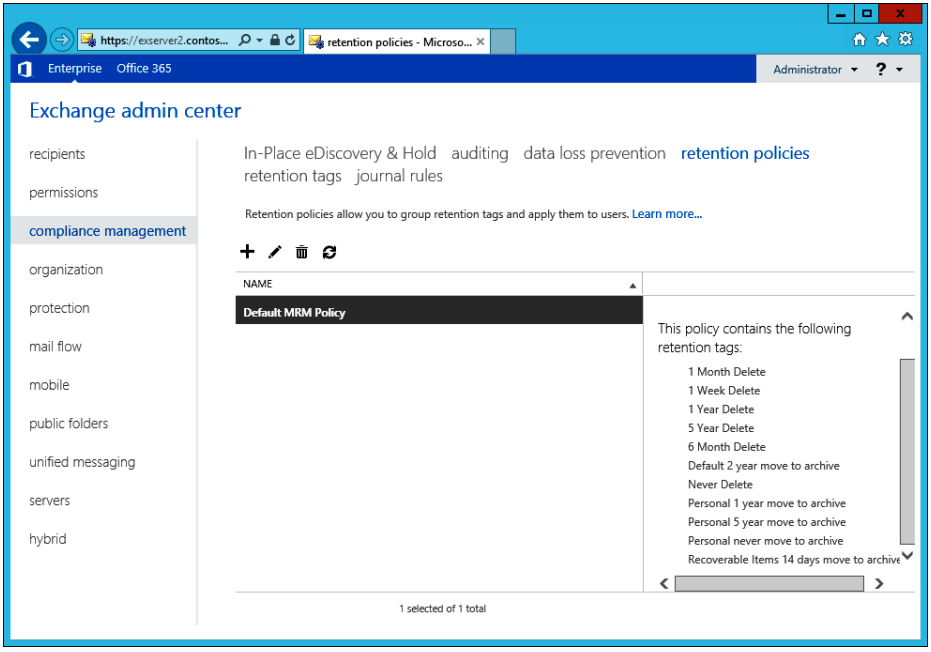


Figure 11-4 The default archive and retention policy Exchange 2013 provides

It might be strange to have multiple default retention policies within an organization. In fact, apart from their name, the policies differ slightly in terms of the retention tags they contain. The Default Archive Policy contains only archive tags and therefore does not function well in terms of an overall retention policy, which ideally should contain tags to help remove unwanted items from a mailbox and archive items that need to be retained. However, although a good reason exists for the change made from Exchange 2010 RTM to Exchange 2010 SP1, the two default retention policies in Exchange 2010 SP1 and Exchange 2013 appear functionally identical, apart from the name. The only reason Microsoft changed the name seems to be to align the default retention policies across Exchange Online and Exchange on-premises because both now use Default MRM Policy.

In any case, the Exchange 2010 SP1 version of the default retention policy will continue to work with Exchange 2013, and there is no reason to replace it with the Exchange 2013 version for the mailboxes that use the older policy. Table 11-1 describes the retention and archive tags that are included in the default policy. You can add retention archive tags to or delete them from the policy if required. You can also see these tags listed in Figure 11-4.

TABLE 11-1 Tags included in default archive and retention policy

Tag name	Type	Purpose
Default 2-year move to archive	Default	Automatically move items to the personal archive when they are two years old. This tag is applied to any item in the mailbox that does not have an explicit tag applied by the user or is inherited when an item moves into a folder that has a default policy.
Personal 1-year move to archive	Personal	Tag that the user can apply to items to instruct the MFA to move the items into the personal archive after they are one year (365 days) old.
Personal 5-year move to archive	Personal	Tag that the user can apply to items to instruct the MFA to move items into the personal archive after they are five years (1,825 days) old.
Personal never move to archive	Personal	Tag that the user can apply to items to block the MFA from ever moving the items into the personal archive.
Recoverable Items 14-days move to archive	Recoverable Items folder	Move items placed in the Recoverable Items folder to the personal archive after 14 days.
1 Month Delete	Personal	Move items into the Deleted Items folder after one month.
1 Week Delete	Personal	Move items into the Deleted Items folder after one week.
6 Month Delete	Personal	Move items into the Deleted Items folder after six months.

1 Year Delete	Personal	Move items into the Deleted Items folder after one year.
5 Year Delete	Personal	Move items into the Deleted Items folder after five years.
Never Delete	Personal	Disabled tag that prevents the MFA from processing the item; the effect is to stop the item from ever being deleted.

Notice that the default MRM policy contains only a single default tag to move items into the archive after two years; no default tag is present to delete items. Therefore, the impact of applying this policy to mailboxes is that every item more than two years old will be moved by MFA into the archive the first time MFA processes the mailboxes after the policy is applied. And because Exchange automatically applies the default MRM policy to mailboxes after they are enabled for archives, this sequence of events becomes immediately apparent if items exist in the mailbox that are more than two years old. In effect, this results in the disappearing-items syndrome in which users log problem reports for missing mailbox items. In the majority of cases, the missing items are found in folders in the archive mailbox. It just takes time for users to realize that Exchange moves items automatically after they reach a certain age; this underlines the importance of communication with the user community as you implement archive mailboxes.

INSIDE OUT **Don't delete the default policy!**

You should not delete the default archive and retention policy because this will affect the processing MFA performs for the mailboxes to which the policy is already assigned. It's a better idea to create a custom archive and retention policy tailored to the needs of the company or to different groups of users and apply that policy to their mailboxes. In this case, the custom retention policy replaces the default archive policy. How to manipulate retention policies and tags is discussed in the "Messaging records management" section later in this chapter.

Using an archive mailbox

Assuming that your mailbox is archive-enabled and a suitable client is at hand, working with items in the archive is just like working with items in the primary mailbox. You can create new items, reply to messages, move items around, and so on. After the archive mailbox is created, it is up to the user to populate it, most likely by using drag and drop to move folders or items from his primary mailbox. Administrators can import the complete

contents of PSTs into a mailbox, but there are some limitations with this approach, as discussed previously.

Exchange doesn't support offline access for data held in archive mailboxes. When Outlook is configured to use cached Exchange mode, it has access only to the offline copies of the folders from the primary mailbox that are stored in the OST and uses background synchronization to keep those folders updated. This arrangement enables Outlook to work through transient network interruptions. Outlook has to be able to connect to the server before it can work with the data stored in an archive.

TROUBLESHOOTING

I can't access my archive mailbox when I'm offline.

If you want something to be available offline (or available to a mobile device), you have to store it in the primary mailbox. The archive is designed to hold information that isn't always required immediately and that you can wait to access until you can get back online, so if you need something from the archive and know that you have to work offline (for example, on a road trip), you have to plan ahead and move the desired items from the personal archive into the primary mailbox beforehand.

Disabling an archive mailbox

If present, you can disable the archive for the selected mailbox in EAC by clicking the Disable link under In-Place Archive shown in the action pane. Alternatively, you can edit the mailbox's properties, open the Mailbox Features section, and disable the archive there. If you have an active EMS session, you can run the Disable-Mailbox cmdlet. For example:

```
Disable-Mailbox –Identity 'Smith, John' –Archive
```

EMS prompts for a confirmation before it proceeds unless you add the –Confirm:$False parameter. This is not a good idea unless you are absolutely sure that you want to disable the archive. If the archive's owner is logged on and connected to her mailbox when the archive is disabled, she can continue to access data in the archive until the next time the client disconnects.

When it disables an archive mailbox, the Information Store disconnects it from the primary mailbox and keeps it in the database until the deleted mailbox retention period expires. If you make a mistake and have to re-enable the archive before it is removed from the database, the Store reconnects the original archive, and all the information in it becomes available to the user again. Note that you cannot disable a mailbox when it is on hold; this action can interfere with the ability to preserve information that might be required

for discovery purposes. If necessary, you can specify the IgnoreLegalHold switch with the Disable-Mailbox cmdlet to force Exchange to disable a mailbox. It would be wise to seek approval from your legal department before taking such an action.

Messaging records management

Exchange 2007 introduced the messaging records management (MRM) system as its business email governance strategy to help users comply with regulatory and legal requirements. The idea is to provide a method for users to retain messages and attachments that are required business records. Another way of thinking about MRM is that it helps users keep control over mailboxes by automating the retention process; marked items are kept as long as required, whereas others can be automatically discarded when their retention period (otherwise known as the expiration limit) expires. The key to success for any scheme that aims to alter user behavior is to make it as simple as possible while achieving maximum functionality. Exchange 2007 didn't quite meet this goal, largely because users did not like being forced to change how they worked. The lack of automated processing spelled disaster for this implementation.

Microsoft therefore needed to change its tactics to provide a workable implementation of MRM. The change occurred in Exchange 2010, and the same approach to messaging management is further developed in Exchange 2013. Managed folders are deprecated in Exchange 2013. At this point, any remaining vestige of managed folders should be eliminated from Exchange deployments as quickly as practicable. See the "Upgrading from managed folders" section later in this chapter.

MRM depends on retention tags that are applied to items in mailboxes through policy and are automatically processed thereafter by the MFA. Retention tags can be applied to any item in any folder to specify what action Exchange should take for the item when its retention period expires. Supported actions include the hard (permanent) or soft (recoverable) deletion of the item, moving the item to a personal archive, or flagging the item for user attention. Retention policies group retention tags together in a convenient manner so administrators can apply policies to mailboxes rather than having to assign individual retention tags to folders. Retention tags and policies are organization-wide objects that are stored in Active Directory and can therefore be applied to any mailbox in the organization after they are created. Just as with Exchange 2007, the MFA is responsible for checking mailbox contents against policy and taking whatever action is determined by policy for items that exceed their retention period.

Types of retention tags

Table 11-2 describes the three types of retention tags Exchange 2010 and Exchange 2013 support. The type shown in the third column is a value passed to the –Type parameter

when you create a new tag with the New-RetentionPolicyTag cmdlet. Exchange uses this value to understand the scope of the items in a user mailbox to which it can apply the tag.

TABLE 11-2 Types of retention tags.

Tag type	Context	Target
Retention policy tags (RPT)	Administrators can apply these tags to default mailbox folders such as the Inbox, Sent Items, and Deleted Items. If an RPT is assigned to a default folder, all items in the folder automatically come under the control of the tag unless the user applies a personal tag to the item. Only one RPT can be assigned per default folder.	Supported for Exchange default mailbox folders such as the Inbox, Calendar, and Sent Items. See *http://technet.microsoft.com/en-us/library/dd297955(v=exchg.150).aspx for an up-to-date list of supported folders.*
Default policy tags (DPT)	A catch-all tag the MFA applies to any item that does not inherit a tag from its parent folder or has not had a tag explicitly applied to it by the user. In other words, if no other tag applies to an item, Exchange will respect the instructions contained in the default tag. A retention policy includes only a single DPT that is used to delete items; you can include another DPT to control the movement of items into the archive. It's logical but sometimes overlooked that if you specify two DPTs in a policy, the tag that moves items into the archive must have a shorter retention period than the tag that deletes items.	*Supported for any folder in a mailbox.*
Personal tags	Users can apply these tags to nondefault folders and individual items in a mailbox. Personal tags that move items into the archive can also be applied to default folders. Personal tags mark an item with an explicit retention, usually to comply with a business requirement. For example, you might use an Audit tag to mark items that users are compelled to retain for audit purposes. A retention policy can include many personal tags.	*Supported for any folder, item, or conversation in a mailbox.*

Microsoft originally restricted the set of default folders to which you could apply a retention policy tag to a smaller set including the Inbox, Sent Items, and Deleted Items. The set has gradually expanded, and you can now define a retention policy tag for just about every default folder, including those such as the Calendar, Contacts, and Tasks, where considerable care must be taken not to interfere with items that users often want to retain for a

considerable time. After all, no one will thank you if you clean out the CEO's calendar after 120 days!

The set of default folders includes those that often accumulate debris within mailboxes. Sync issues, Junk E-Mail, and RSS Feeds are particularly interesting in this respect. It's good to have these folders cleaned out automatically because the items stored here aren't typically needed after a day or so.

When you create a new retention tag with EAC, you select the type of tag through a drop-down list (Figure 11-5).

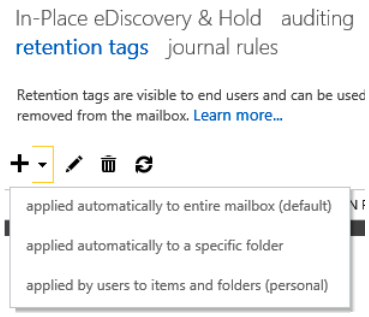


Figure 11-5 The EAC drop-down list for retention tags

The three values are:

- Applied automatically to entire mailbox This is a default policy tag (DPT). You can have two of these in a policy, one for deleting items, the other for archiving items. As pointed out in Table 11-2, if you use two default tags, the default tag used to archive items must have a shorter retention period than the tag that deletes them. Exchange applies default tags to any untagged item in a mailbox. Untagged items are those that do not inherit a tag based on the folder in which they are stored or have not had a tag placed on them by a user. Because of its influence over all untagged items in a mailbox, the default tags are critical in terms of how long items remain in a mailbox before they are deleted or archived.

- Applied automatically to a specific folder This refers to retention policy tags (RPTs) that are associated with one of the supported default folders such as the Inbox, Sent Items, and so on. You can have as many RPTs as you like for a default folder, but only one RPT for a folder can be included in a policy. EAC signals the error shown in Figure 11-6 if you attempt to add two RPTs for a default folder to a retention policy.

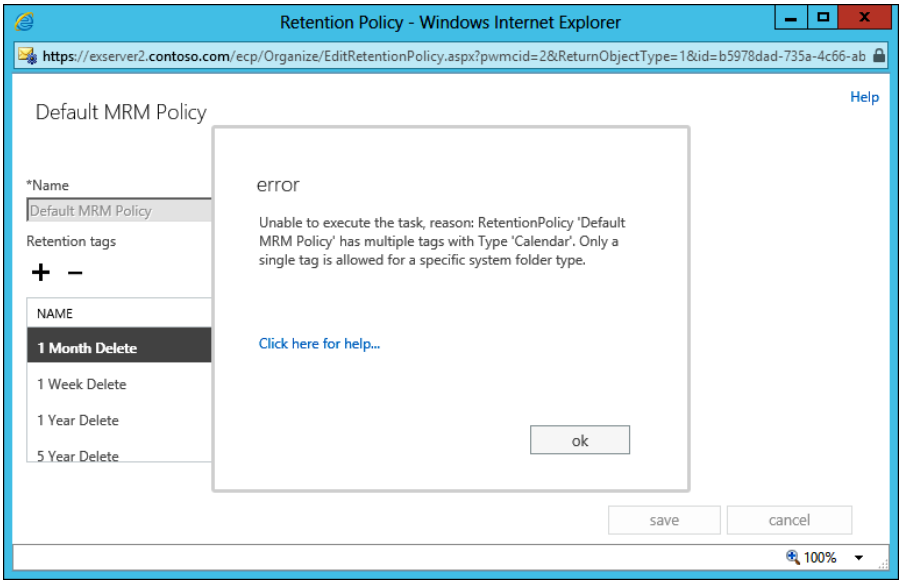


Figure 11-6 Error when attempting to add two RPTs for the same folder to a policy

- Applied by users to items and folders (personal) As the name implies, these tags are placed on folders or individual items as a result of users making a personal decision that these contain information that needs to be retained for some out-of-norm period. For instance, a folder containing items required for financial audits might need to be retained for six years and then deleted. Users could accomplish this goal by placing a personal tag with a retention period of six years and a retention action of *DeleteAndAllowRecovery* on the folder that holds the items required for audit purposes. Other items in the mailbox that are not in the folder can also be tagged with the personal tag to retain them for six years.

The range for a retention period is from 1 day to 24,855 days or, roughly, 68 years. Probably no one reading this book will worry whether an item that is 68 years old is deleted or moved in an archive, but it is good to know that such an extended period is feasible. Or maybe it's just a question of Exchange trivia.

INSIDE OUT Some items are timeless

Items in some folders tend to be more timeless than general-purpose messages, so you should think carefully through the potential consequences when you create retention policy tags for folders. For example, contacts tend not to expire, and people usually want to keep them for a long time, so it might be best to create retention policy tags that enable users to mark items not to expire. To do this, they should create a personal tag with a Never retention period, which indicates to Exchange that the item should be held indefinitely and neither deleted nor archived by MFA.

INSIDE OUT Creating a default tag for voice mail

Although I have said that you can have only two default tags in a retention policy (one to archive items, one to remove them), Exchange allows a special condition for voice mail. You cannot create a default policy tag for voice mail through EAC. Instead, you run the New-RetentionPolicyTag cmdlet through EMS to create the new default tag. After it's created, the new tag appears in EAC and can be included in retention policies. Here's an example of creating a new default tag that removes voice mail after 14 days.

```
New-RetentionPolicyTag -Name "Voicemail 14 days" -Type All -MessageClass
Voicemail -AgeLimitForRetention 14 -RetentionAction DeleteAndAllowRecovery
```

Retention tags cannot be applied to items directly. First, they have to be assigned to a retention policy and the retention policy assigned, in turn, to the mailboxes whose content you want to manage. A retention tag can be reused several times in different policies. Although there is no theoretical limit to the number of retention tags you can define for an organization, it makes sense to create a set of tags that can be shared and reused between retention policies rather than creating separate tags for each policy.

Exchange can apply only one retention policy tag and one archive tag to an item. Two simple rules are enforced when Exchange evaluates policies that it can apply to an item. The first rule states that the policy with the longest retention period always wins and is intended to ensure that Exchange never deletes an item before its time truly expires. The second rule is that an explicit policy is always respected before an implicit or default policy. If you apply a personal tag to an item to retain it for six years and the default retention policy for the folder requires deletion after 12 months, the item will be kept for six years. Personal tags

can be placed on items, conversations, or complete folders, and they are transferred with items if you move them between folders.

> **Note**
>
> **When you apply a tag to a conversation, you really just apply the tag to the items that make up the conversation at that point in time. Exchange knows that the items are part of the conversation and can apply the tag, but it won't look for and tag new items as they arrive and join the conversation. This is because a conversation is not a real storage container within a mailbox and therefore cannot be permanently tagged. In short, tags only exist in a persistent manner for folders and individual items.**

Of course, to make any sense of retention policies, you also need to deploy clients that include the necessary intelligence and user interface. The only clients in this category are Outlook 2010, Outlook 2013, and Outlook Web App. At the time of writing, no mobile client has any ability to display or set retention tags (this situation might change with updates to the Outlook Web App for Devices app). As you'll see when you review how retention policies function from a user perspective, the Outlook user interface provides the richest views of retention policies and tags. Outlook Web App is less capable but still highly usable.

System tags

Exchange 2013 supports two types of retention tags: system tags and nonsystem tags. Exchange uses system tags for its own purposes, and they are not shown when you run the Get-RetentionPolicyTag cmdlet unless you specify the –IncludeSystemTags parameter. By default, Get-RetentionPolicyTag lists only nonsystem tags (those created to be used with normal retention policies). To see the system tags defined in an organization, you can execute the following command. (Nonsystem tags are listed afterward.)

```
Get-RetentionPolicyTag –IncludeSystemTags | Format-Table Name, Type, SystemTag
```

The first three entries you will see (AutoGroup, ModeratedRecipients, and AsyncOperationNotification) are system tags that Exchange uses to prevent items from accumulating in arbitration mailboxes. The other entries are nonsystem tags, which instruct the MFA to clean out these mailboxes as items expire. To see details of the retention policy used for arbitration mailboxes and its links to the two system tags, run these commands:

```
Get-RetentionPolicy –Identity 'ArbitrationMailbox'
Get-RetentionPolicyTag –Identity 'AutoGroup'
Get-RetentionPolicyTag –Identity 'ModeratedRecipients'
Get-RetentionPolicyTag –Identity 'AsyncOperationNotification'
```

CAUTION!

You cannot add system tags to a retention policy that's applied to user mailboxes. Deleting a system tag is also a bad thing because you have no idea of what consequences might follow from this event.

Designing a retention policy

Many retention policy tags can exist within an organization. This allows great flexibility in creating appropriate policies for different groups that work within a company. For example, the finance department might want Exchange to delete everything in the Deleted Items folder permanently that is more than three days old (the shred principle), whereas users in other departments might not be concerned whether items survive in the Deleted Items folder for 30 days or more. You can apply a retention policy to members of the finance department that includes a retention policy tag for the Deleted Items folder that instructs the MFA to remove items after three days. The same policy might include a personal tag that allows members of the finance department to mark items that have to be archived for audit purposes after a month in the primary mailbox. The MFA moves items with this tag to the archive mailbox when it processes the mailbox.

Why are you creating this retention policy?

Before you rush to create a retention policy for anyone—even the finance department—you should sit down and determine the why, when, and how for the policy:

- **Why you are implementing the policy? What business need will the policy serve, and how does the policy contribute to the governance of information required by the business? How will the policy assist the business in satisfying its legal and regulatory requirements?**
- **When will you implement the policy? To what mailboxes will the policy be applied? How will you communicate the policy to end users so that they understand the purpose of the policy and how it will affect the contents of their mailboxes?**
- **How will you implement the policy? What tags and types of tags are required? What actions will you enforce through tags, and what retention periods are used? Do any restrictions exist as a result of other aspects of your deployment? For example, if you use an archiving product from another vendor, you cannot deploy tags to move items into an archive mailbox after a designated period.**

The design for a retention policy might be captured in a simple table format that makes it clear which tags are included in the policy, their purpose, and the folders MFA processes. Apart from its other advantages, capturing the design like this makes it easier to communicate the policy to users. Table 11-3 lays out a simple policy that could be applied to help managers cope with overloaded mailboxes.

TABLE 11-3 Laying out a retention policy

Retention Policy Name	***Management Retention Policy***		
Applies to	***Mailboxes with CustomAttribute7 = Management***		
General purpose	Automatic clean-out of Inbox and Sent Items folders to encourage users to keep these folders tidy. Items in most other folders can remain in place for a year, but calendar items are kept for five years. Removal of items from the Deleted Items folder after a week. Tags are provided to keep the contents of several utility folders under control. A tag is provided to enable users to mark items for retention for 10 years. Another is available to tag audit items so that they are held indefinitely.		
TAG NAME	**TAG TYPE**	**APPLIES TO**	**ACTION**
Inbox 30	RPT	Inbox folder	Move items to Recoverable Items after 30 days.
Sent Items 30	RPT	Sent Items	Move items to Recoverable Items after 30 days.
Calendar 5 years	RPT	Calendar	Move calendar items to Recoverable Items after five years.
Deleted Items 7	RPT	Deleted Items	Permanently remove items after seven days.
Junk Mail 3	RPT	Junk Mail	Permanently remove items after three days.
RSS Feeds 3	RPT	RSS Feeds	Move items to Recoverable Items after three days.
Sync Issues 1	RPT	Sync Issues	Clean out synchronization error logs daily.
Delete after 5 years	DPT	All folders	Move items to Recoverable Items after 1825 days.
Archive after 2 years	DPT	All folders	Move items into the archive mailbox after 730 days.
Retain for 10 years	PER	All folders	Move items to Recoverable Items after 3650 days (10 years).
Keep for Audit	PER	All folders	Never delete—item held indefinitely for audit purposes.

Logically, you can have only a single RPT for each default folder within a retention policy. It would be very confusing to have two retention policies compete within a single folder! In addition, a retention policy can have only one default retention tag and one default archive policy tag that apply to all folders. If you use two default tags (which implies that you use

archive mailboxes), you should make sure that items can move into the archive before they are deleted. This is done by giving the default archive tag a shorter retention period than the default retention tag. For example, this will work:

- **Default archive tag retention period 2 years (730 days)** MFA will move all items that do not have another tag on them into the archive after items are more than two years old.
- **Default retention tag period 5 years (1,825 days)** MFA will remove the items (either allowing recovery or removing them permanently) after they have been in the mailbox for five years. In fact, the items were in the primary mailbox for two years, then moved to the archive, and then aged out of the archive after another three years.

Things would not run so smoothly if the two retention periods were reversed, because MFA would then delete items before they had a chance to be archived!

INSIDE OUT **Keep it simple**

Exchange enables you to create and apply as many retention policies as you want, but the question of long-term supportability arises. You should also consider the question of how many retention policies are really required for the organization as a whole and attempt to restrict the number to the minimum necessary to meet business needs. A couple of well-designed, logical policies that satisfy the vast bulk of requirements will be easier to create, deploy, and manage on an ongoing basis than a mass of granular policies generated to meet the specific needs of a department or other business group that might disappear following the next corporate reorganization. The more policies exist, the more potential there is to confuse administrators and users alike.

Managed Folder Assistant and retention policies

The MFA is responsible for implementing the actions specified in retention and archive tags when it processes a mailbox. For example, if the retention period for the Inbox is 30 days, the MFA will tag any item aged up to 30 days and take the specified action for items aged 30 days. Therefore, before you implement a policy that could affect thousands of items in user mailboxes, it is critical to communicate clearly what will happen, when it will happen, and how users can prepare for the implementation of the retention policy and respond to its actions afterward. You might have to communicate several times before the retention policies are implemented to avoid a deluge of calls to the help desk the morning after the

MFA runs. Having a well thought-out retention policy that the business agrees on and that is then clearly communicated to business leaders and end users well in advance is a good way to avoid problems when you implement this part of your compliance strategy.

Exchange uses the date and time when an item is created in a user's mailbox as the baseline to calculate the age of the item for retention purposes, so an age limit of 30 days for the Inbox default retention tag means that items become eligible for processing by the MFA 30 days after they are delivered to the Inbox. The creation date is used for retention purposes even for modifiable items such as posts. As noted earlier, you can create a tag to mark items never to be processed by the MFA. Such a tag has no value set for its AgeLimitForRetention property, and its RetentionEnabled property will be set to $False. For example, to see the set of retention tags that can be added to retention policies to allow users keep items indefinitely, you can run this command:

```
Get-RetentionPolicyTag | Where-Object {$_.AgeLimitForRetention –eq $null} |
Format-Table Name, AgeLimitForRetention, RetentionEnabled -AutoSize
```

Tags with a null age limit might have different retention actions (move to archive or delete), but in effect, they are simply different names for the same effect, which is to tell MFA to ignore them when it processes items. Having tags that do the same thing with different names is generally frowned upon because it increases the number of tags that have to be managed and can cause a little confusion, but sometimes it is justified when you want to communicate to the user the exact nature of the tag and what it will do. Keep For Audit is a good example here; the user should be under no illusion about what this tag means and what it should be used for.

Naming retention tags

When we started to work with retention policies and tags in Exchange 2010, the initial advice given from the Exchange development group was that tags should be named in a way that made their purpose immediately obvious. The idea was that you could use a prefix of RPT for retention policy tags, DPT for default policy tags, and PER for personal tags. Following a naming scheme like the one outlined earlier is logical, but it runs contrary to the way human beings absorb information. Having a tag called RPT-Inbox-30 makes perfect sense to those who are used to interpreting cryptic names that include acronyms. Thus, an administrator who is accustomed to working with retention tags understands at a glance that the name means the tag is a retention policy tag for the Inbox that likely sets a retention period of 30 days.

Client user interfaces expose the names of tags to end users. Outlook Web App exposes less than Outlook does, but the fact remains that users can become aware of tag names. It is for this reason that you need to pay attention to the names given to tags and make sure that the names are logical and convey the meaning of the tag to users.

You must determine a tag naming scheme because the retention policy menu that Outlook 2010, Outlook 2013, and Outlook Web App display lists tag names and their retention period (such as six months) to end users but doesn't display any other detail, such as the action that will be taken when the tag's retention period expires. This is apparent from the screen on the left in Figure 11-7, which is revealed when a user right-clicks an item. (We know the item is in the Inbox because the folder policy is Inbox 30.) Tags can have a variety of associated actions, from permanent deletion to merely warning that the retention period has expired, but the action the tag invokes is not immediately apparent here. Outlook Web App displays retention tags in much the same way. Logically, archive policy tags appear only if a mailbox has an archive.

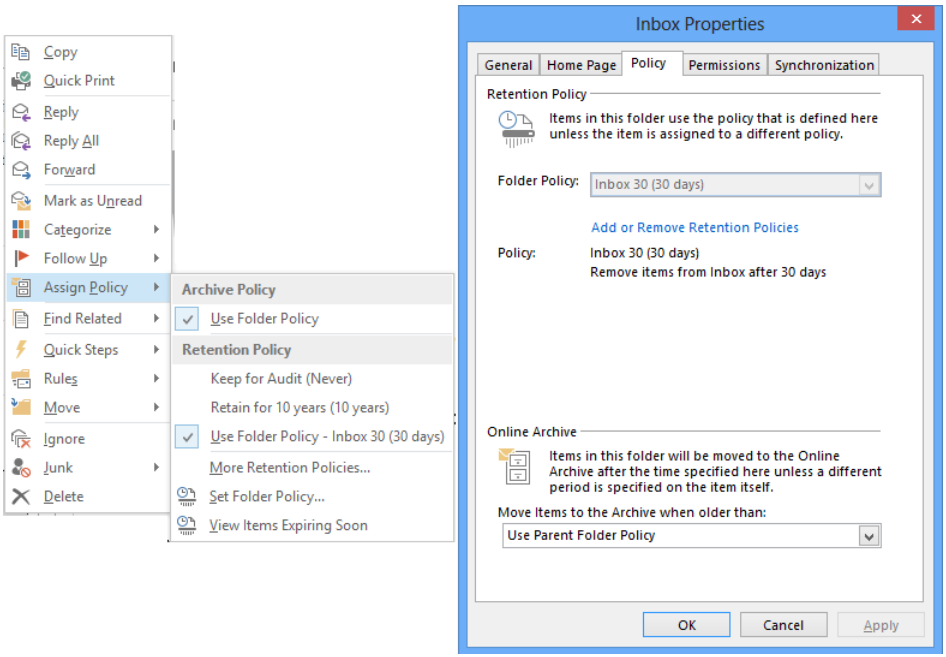


Figure 11-7 Available retention tags that Outlook 2013 lists

Outlook 2010 and Outlook 2013 users can view the actions for the default tag (if one exists) for a folder by viewing the folder properties (the right-side screen in Figure 11-7). Here you can see the action dictated by the tag, in this case to remove items from the Inbox after 30 days. However, this information is not available through the Outlook Web App user interface, and nothing at all is exposed about retention tags in the Outlook 2007 user interface. It can be argued that the tags used in an archive policy and displayed in the archive menu are an exception because users should know that the purpose of these tags is to move items into the personal archive when their retention period expires, but that's still no reason to use cryptic tag names.

The question, therefore, has to be asked whether you should use a more user-friendly naming scheme for retention tags. For example, would RPT-Inbox be better named Inbox Retention Policy, and should PER-Retain be called Retain For Five Years? Some prefer the structure of the first approach, but users usually find the second approach easier to understand.

Another approach that is often taken is to use names that give clear business directives for retention tags. For example, you might use names such as these:

- Business Critical
- Partner Negotiations
- Legal Retention

Tags named like this are usually more specific to departments or groups than more generic names such as Keep For Five Years or Required For Annual Audit, so you might have to define a set of retention tags for each department to match its work practices. The Keep For Audit personal tag specified in the prototype retention policy discussed earlier is an example of a business-specific tag name.

It's *impossible* to give a definitive answer about a naming convention that is suitable for all deployments. Some organizations are happy with cryptic tags because they are a standard that is valid no matter what language is used to connect to Exchange; others will elect to use more user-friendly names because it's easier to communicate the purpose of a retention policy to users, and they feel that this will both ease the introduction of retention policies within the organization and avoid some calls to the help desk. The important thing is to make a decision before you start to design and implement retention policies because changing the names of tags halfway through a deployment is guaranteed to cause maximum confusion.

Creating retention tags

Retention policies and tags are managed through the Compliance Management section of EAC. To create a new retention tag, click Retention Tags and then click New (+). You then select the type of tag (default, folder, or personal) you want to create. In this example, you create a tag for the Inbox folder. Figure 11-8 shows the fields you need to complete, split across two screens.

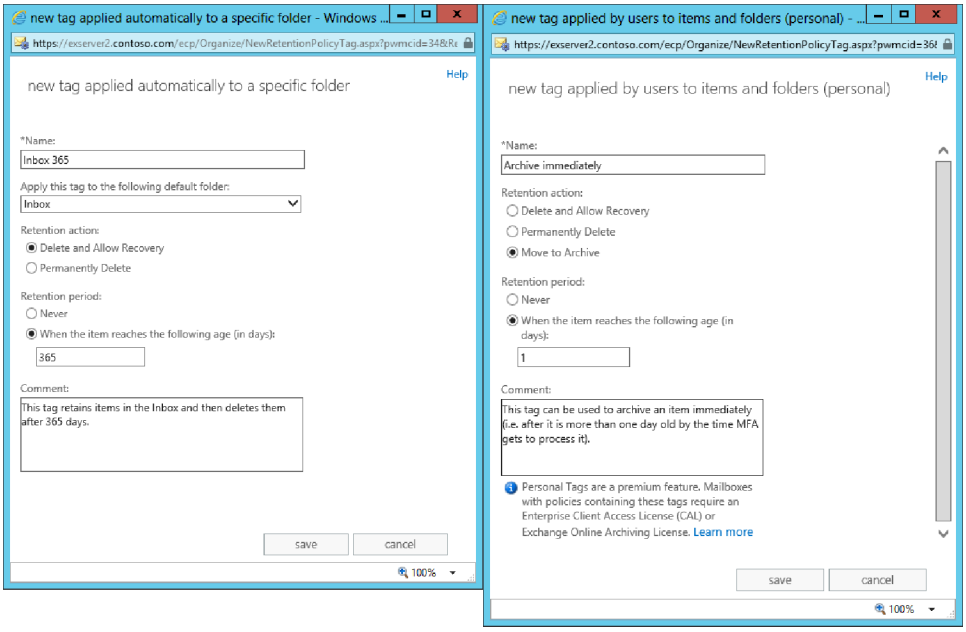


Figure 11-8 Creating a new retention policy tag

A folder tag does not offer you the opportunity to move the item into the archive because Exchange limits this option to a default tag that applies to all untagged items in a mailbox or a personal tag that a user can apply himself. If you were to create a new personal tag, for instance, EAC displays a screen similar to that shown in the right side of Figure 11-8, and you can see that Move To Archive is now available.

After creating all the retention policy tags you need, you should have something like the situation illustrated in Figure 11-9. This is the complete set of the retention policy tags defined for the organization. You can immediately see the advantage of following a well thought-out naming convention for tags. In addition, you can see how it is possible to accumulate a relatively large number of tags quickly, which different retention policies use in an organization. With some forethought, it is possible to reduce the total number of tags by designing some utility tags that are included in every policy, which then means that the only additional tags you need to define are those specifically required by a policy. For example, you can probably define utility tags to clean out folders such as Junk E-Mail and RSS Feeds that apply the same retention period and action for every policy. You might not be as successful in defining utility tags for default folders such as the Inbox or Sent Items because different sets of users might need to keep items in these folders for different periods.

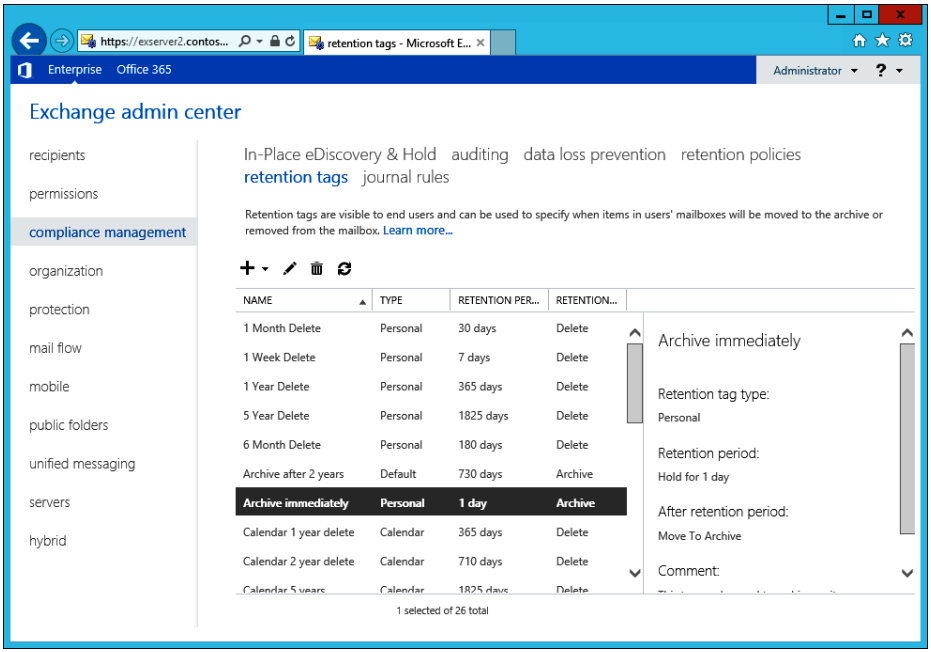


Figure 11-9 Viewing the set of retention policy tags defined for the organization

As an example of how to accomplish the same task with EMS, here is how to create the retention policy tag defined in the management retention policy for the Inbox folder.

```
New-RetentionPolicyTag –Name 'Inbox 30'  –RetentionAction DeleteAndAllowRecovery
–AgeLimitForRetention 30 –Type Inbox –Comment 'Inbox items are automatically deleted
after 30 days' –RetentionEnabled $True
```

You can check the properties of your new retention tag with the Get-RetentionPolicyTag cmdlet. For example:

```
Get-RetentionPolicyTag -Identity 'Inbox 30' | Format-List
```

You should see that the output from Get-RetentionPolicyTag confirms that the tag can cover any class of item (MessageClass = *, the default for Exchange 2010 and Exchange 2013), that it is for the Inbox folder (Type = Inbox), and that items that inherit this tag because they are stored in the Inbox folder will be moved to the Deleted Items folder after 30 days (indicated in the RetentionAction and AgeLimitForRetention properties). In fact, unlike managed folders, retention tags don't accommodate the notion of item segregation; you cannot build a retention tag that applies only to items of a certain class in a folder (such as applying the policy to items of class IPM.Note but ignoring those of class IPM.Contact).

Now review the EMS code that could be used to create some of the other tags that are required for the management retention policy. This time, a personal tag (Type = Personal) is needed to enable users to mark items to be kept in their mailbox for 10 years (3,650 days, give or take a couple of extra days for leap years), after which the items will be automatically moved into the Recoverable Items folder. Exchange gives the action and retention period defined in a personal tag priority if a user applies it to an item in a folder that's already under the control of a retention policy tag. If a user applies this tag to an item in the Inbox, Exchange will not delete it after 30 days as called for by the retention policy tag associated with the Inbox. Instead, Exchange will respect the action and retention period defined in the personal tag because the rule is that an explicit policy always trumps an implicit policy. In addition, remember that Exchange keeps a personal tag on an item even if the item moves to another folder that has an associated retention policy tag. You create the new personal tag with the following command:

```
New-RetentionPolicyTag –Name 'Retain for 10 years'' –RetentionAction
PermanentlyDelete –RetentionEnabled $True –AgeLimitForRetention 3650 –Type Personal
–Comment 'Item to be kept for ten years before it permanently deleted'
```

Setting the Type parameter to Personal is critical here because it makes the tag personal and explicit rather than implicit, like the tags applied to all items in a folder.

TROUBLESHOOTING

I created a retention tag with the wrong type. What do I do?

If you make a mistake and create a retention tag of the wrong type, you cannot change the type with the Set-RetentionPolicyTag cmdlet. Instead, you must delete the tag with the Remove-RetentionPolicyTag cmdlet (or through EAC) and then re-create it afterward with the correct type. The sooner you realize the mistake, the better because this will restrict the processing that MFA does to apply and then remove the tag from items in user mailboxes.

To complete the design, the policy needs to provide managers with two default tags. The first forces any items older than two years (730 days) to be moved into the archive; the second removes items to the Recoverable Items folder after they are five years (1,825 days) old. Remember, a default tag is used when no other tag has been applied to an item.

```
New-RetentionPolicyTag –Name 'Archive after 2 years' –RetentionAction MoveToArchive
–RetentionEnabled $True –AgeLimitForRetention 730 –Type All –Comment 'Items older
than two years are moved to the archive unless otherwise tagged'
New-RetentionPolicyTag –Name 'Delete after 5 years' –RetentionAction
DeleteAndAllowRecovery –RetentionEnabled $True –AgeLimitForRetention 1825 –Type All
–Comment 'Items older than five years are removed from the mailbox'
```

You'll have noticed that all the tags you created specify –RetentionEnabled $True. Therefore, the tag is active and should be processed by the MFA. To disable a tag, set the RetentionEnabled property to $False. The MFA ignores a tag in this state.

Most of the tags you have created so far use the DeleteAndAllowRecovery action, with the exception being the default archive tag. The full set of available actions is as follows:

- **DeleteAndAllowRecovery** Moves the item into the Recoverable Items folder so that the user can recover it if necessary.
- **PermanentlyDelete** Immediately deletes the item so that it cannot be seen using Recover Deleted Items. If the mailbox is on retention hold or subject to an in-place hold, the item is retained in the Recoverable Items folder and still available in eDiscovery searches.
- **MoveToArchive** Moves the item to a folder of the same name in an archive mailbox. This action is only possible if the mailbox has an archive. If not, the action is ignored. Moving to the archive is analogous to the Outlook AutoArchive option that moves items into a PST on a regular schedule to help keep a mailbox under quota. The big difference is that users can't vote whether they want to use the option because Exchange moves items into the personal archive automatically without asking for user opinion. Policies that move items into an archive mailbox are known as archive policies. Exchange ignores the archive tags if you create a retention policy that includes tags to move items into the archive and apply it to a mailbox that doesn't have a personal archive. If the mailbox is subsequently assigned a personal archive, the MFA applies the archive tags for the mailbox the next time it runs.

Tags that specify MoveToArchive are sometimes called move tags, whereas those that specify DeleteAndAllowRecovery and PermanentlyDelete are called delete tags. To check that all the required tags are in place that you need to build the retention policy, you can review the set of tags through EAC or execute the following EMS command:

```
Get-RetentionPolicyTag | Format-Table Name, Type, RetentionAction, RetentionEnabled,
AgeLimitForRetention –AutoSize
```

If you make a mistake in defining a retention policy tag, you can remove the tag with the Remove-RetentionPolicyTag cmdlet. If the tag has already been applied to mailbox items, the MFA cleans up by removing any reference to the removed tag from items as it processes mailboxes.

INSIDE OUT **An advantage of using New-RetentionPolicyTag**

EAC offers administrators the choice of three actions to assign to a retention tag. They can delete and allow recovery, permanently delete, or move to the archive. In fact, Exchange supports a fourth action, which is to mark the item as expired. An expired item simply means that Outlook or Outlook Web App displays an indication to the user that the item is past its best-by date, so it is less destructive in some respects than removing or archiving items. MFA eventually removes or archives expired items if default tags exist in the policy to mandate these actions, but in the meantime, it's up to the user to decide whether to keep the item (in which case she should apply a new retention policy to it) or delete it. Marking items as expired is often used to test the effect of a retention tag on items because it is less disruptive than a tag that deletes or archives an item. Here's an example of how to create a retention tag for the Inbox that expires items after 30 days:

```
New-RetentionPolicyTag –Name 'Inbox Expiry' –RetentionAction
MarkAsPastRetentionLimit –RetentionEnabled $True –AgeLimitForRetention 30 –Type
Inbox –Comment 'Anything in the Inbox more than 30 days is marked as expired'
```

Creating a retention policy

Now that you have created the necessary retention tags to help managers impose order on their mailboxes, you can create a new retention policy. Under the Compliance Management section of EAC, choose Retention Policies and click New (+). You have to name the retention policy and add the set of retention tags that enable the new policy to impose some level of retention compliance on the users to whom you assign the policy.

Retention policy names are entirely internal and are never revealed to users. The major step in building the policy is to scan the set of available tags and select the necessary ones (Figure 11-10). When the required tags are selected, click OK to return to the initial screen and then Save to create the new policy. EAC then checks to ensure that the set of tags is valid and that you haven't done something such as specifying two default retention tags or two folder tags for the same folder.

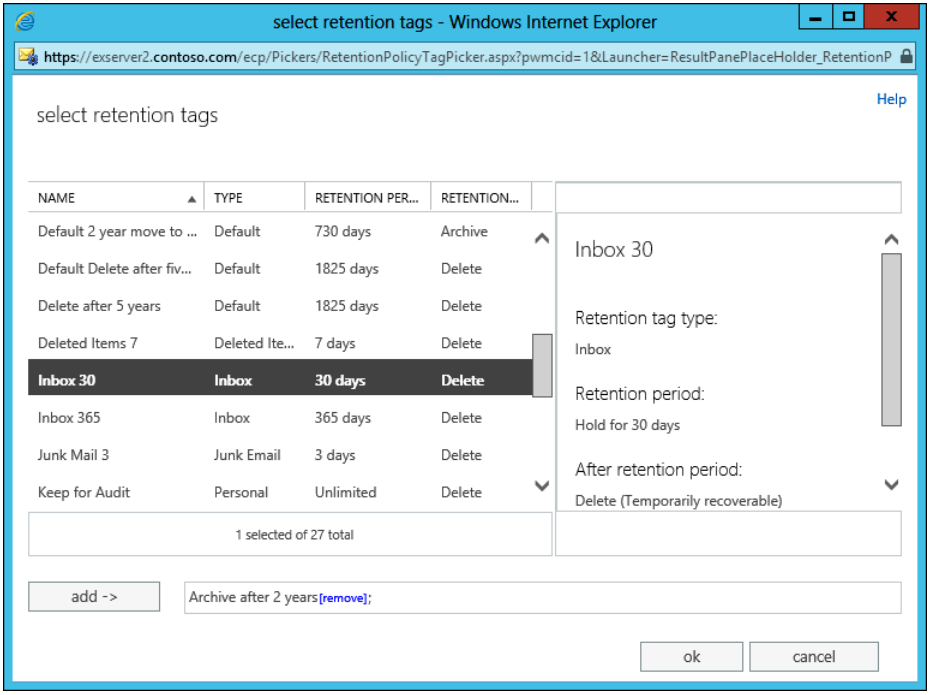


Figure 11-10 Adding retention tags to a retention policy

When EAC returns to Compliance Management, you should then see something like Figure 11-11 with the new policy listed in the set of retention policies defined for the organization and the set of retention tags used by the policy shown in the action pane.

You can also create retention policies through EMS by using the New-RetentionPolicy cmdlet. In this command, you create the policy and associate the 11 tags you want to use with the new policy. Don't worry about uppercasing the tag names as defined. EMS can locate tags as long as you spell their names correctly.

```
New-RetentionPolicy –Name 'Management Retention Policy' –RetentionPolicyTagLinks
'Archive after 2 years', 'Calendar 5 years', 'Delete after 5 years', 'Deleted Items
7', 'Inbox 30', 'Junk Mail 3', 'Keep for Audit', 'Retain for 10 years', 'RSS Feeds
3', 'Sent Items 30', 'Sync Issues 1'
```

You can examine details of the new retention policy with the Get-RetentionPolicy cmdlet:

```
Get-RetentionPolicy –Identity 'Management retention policy' | Format-List
```

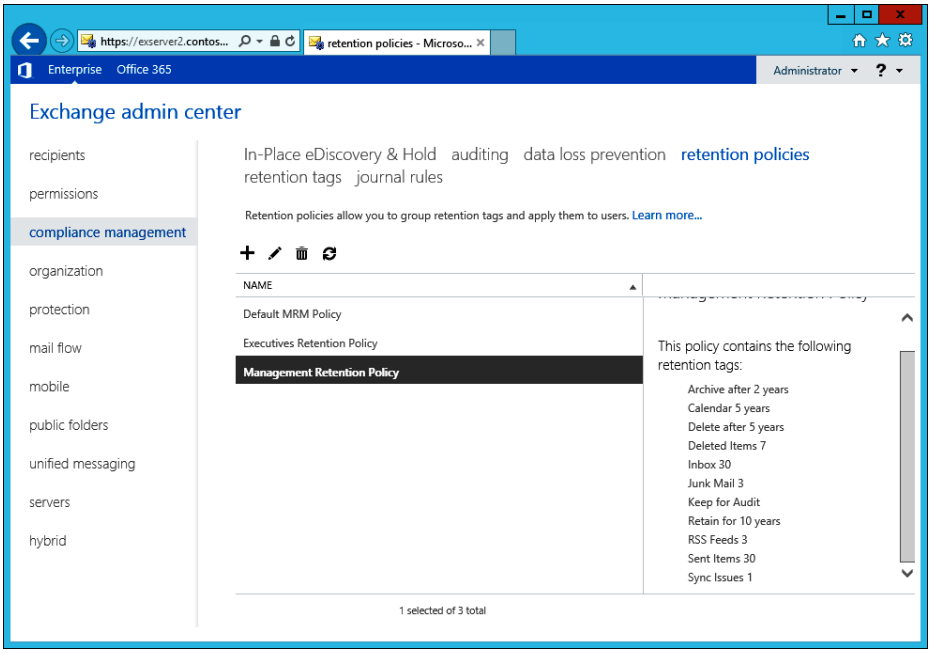


Figure 11-11 Reviewing the tags in a new retention policy

Some of the properties Get-RetentionPolicy reveals apply to Exchange Online only. For example, Exchange Online enables administrators to mark a retention policy as default (IsDefault = $True), meaning that Exchange applies this policy to users' mailboxes if another retention policy isn't assigned.

Provided there is good reason for each tag to exist in the policy, a 10-tag policy composed of some folder tags, a default tag, and some personal tags is a reasonably simple retention policy because other policies can incorporate a lot more tags to create a very exact retention environment for a user to operate within. You might have more tags than this if you decide to include a retention policy tag for every default folder. Using retention policy tags to clean out items that otherwise accumulate and are never cleared out in default folders such as Sync Issues, Junk E-Mail, and RSS Feeds is a good example of where you can gain real value from a well-designed retention policy.

INSIDE OUT Good reasons to limit the number of tags in a policy

Microsoft recommends that you have no more than 10 personal tags in a policy; otherwise, you might confuse users with too much choice. (Only personal tags usually show up in client user interfaces, hence the focus on these tags.) This is reasonable advice, but as with most advice, there will be times when you need to incorporate more tags in a policy to meet specialized business needs. A more sophisticated policy for a department might have separate retention tags for many of the default folders, a set of personal tags developed specifically to suit the retention needs of the department, and a default retention tag for everything else. User interface constraints are another good reason for limiting the number of tags in a policy. If you have five personal tags or fewer in a policy, there's a reasonable guarantee that Outlook Web App can display all the tags in its user interface. However, if you include 20 personal tags in a policy, you'll find that Outlook Web App cannot list all the tags, and some will simply drop off the end of the available list. This is especially true when you consider the relatively small display screen of some of the devices that Outlook Web App now supports.

Outlook 2013 takes a different approach from Outlook Web App by displaying up to 10 personal tags when a user selects an item, right-clicks, and selects Assign Policy from the shortcut menu. This list includes an entry for More Retention Policies that reveals the full set of personal tags defined in the policy. As you can see from the rightmost screen in Figure 11-12, the full set of retention tags for the policy assigned to a mailbox contains 14 tags, many of which are of dubious value. Using a separate dialog box keeps the Outlook user interface under control while still permitting users to discover all the tags available in a policy at the expense of requiring separate clicks and additional knowledge of where to find the tags. For all these reasons, it's just a bad idea to create a tag-filled policy.

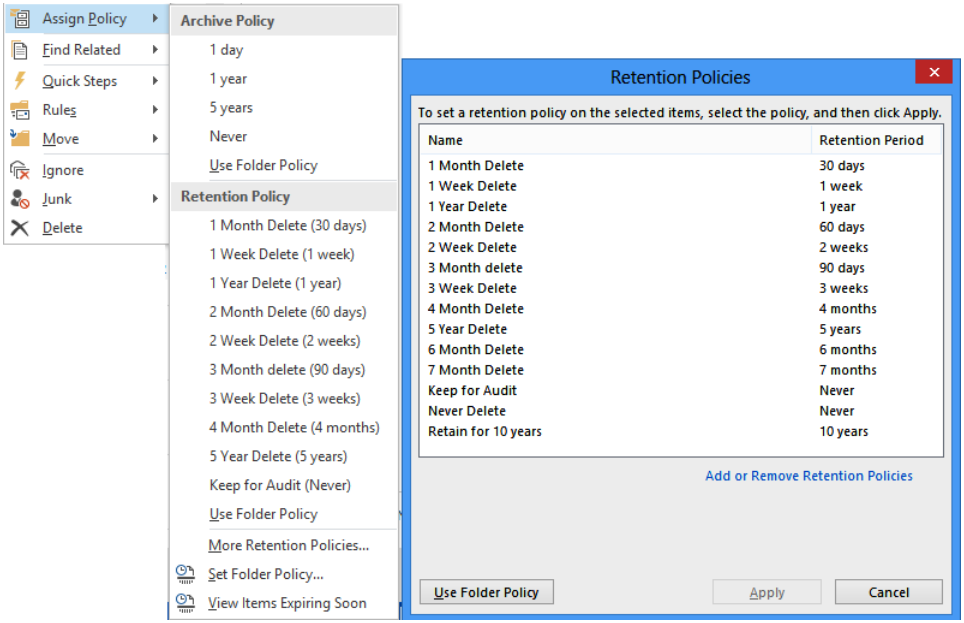


Figure 11-12 How Outlook 2013 displays the full set of personal tags to users

INSIDE OUT Calendar and Tasks

Although retention policies support tags for calendar and task items, no client currently exposes the necessary user interface to enable users to apply policies to these folders. In some respects, this is a good thing because it means that administrators are the only people who can set a retention policy for these important folders. Some might disagree because they want to apply a different retention policy, but that's the current situation.

Applying a retention policy to mailboxes

You can apply retention policies to mailboxes by:

- Selecting mailboxes individually in EAC and editing the mailbox properties to assign a retention policy. This option is acceptable when you have to deal with only a few mailboxes. You cannot set a retention policy on a mailbox when it is initially created with EAC; this has to be done afterward, possibly because the use of personal tags

in a retention policy requires an enterprise CAL. The standard CAL permits the use of default and folder tags in a retention policy.

- Selecting multiple mailboxes and applying a Bulk Edit. This exposes options available for all the selected mailboxes in the action pane. The option to enable an archive for all the selected mailboxes is available through the More Options link at the bottom of the action pane.
- Running the Set-Mailbox cmdlet to apply retention policies to a group of selected mailboxes. This is clearly a more efficient approach to take when you have to deal with more than a few mailboxes at one time.

When assigned to a mailbox, the policy becomes active the next time the MFA processes the mailbox. At this point, MFA writes the policy information into the mailbox. (See the "Behind the scenes with the MFA" section later in this chapter.) Until MFA processes the mailbox, the policy tags will not appear in the client user interface, so the lack of these tags is a good indication that the MFA has not yet gotten to a mailbox. You can force the issue by running the Start-ManagedFolderAssistant cmdlet as follows, passing the name of the mailbox you want MFA to process:

```
Start-ManagedFolderAssistant –Identity 'Tony Redmond'
```

If you're setting a policy for a group of users (see Figure 11-13), you'll probably do it in one operation by selecting the mailboxes with the Get-Mailbox cmdlet and piping the results to Set-Mailbox. For example:

```
Get-Mailbox –Filter {CustomAttribute7 –eq 'Management'} | Set-Mailbox
–RetentionPolicy 'Management retention policy' –RetentionComment 'Management
retention policy applies to this mailbox'
```

To discover the set of mailboxes that have retention policies in place, you can use a command like this:

```
Get-Mailbox –Filter {RetentionPolicy –ne $Null} | Format-Table Name, RetentionPolicy
–AutoSize
```

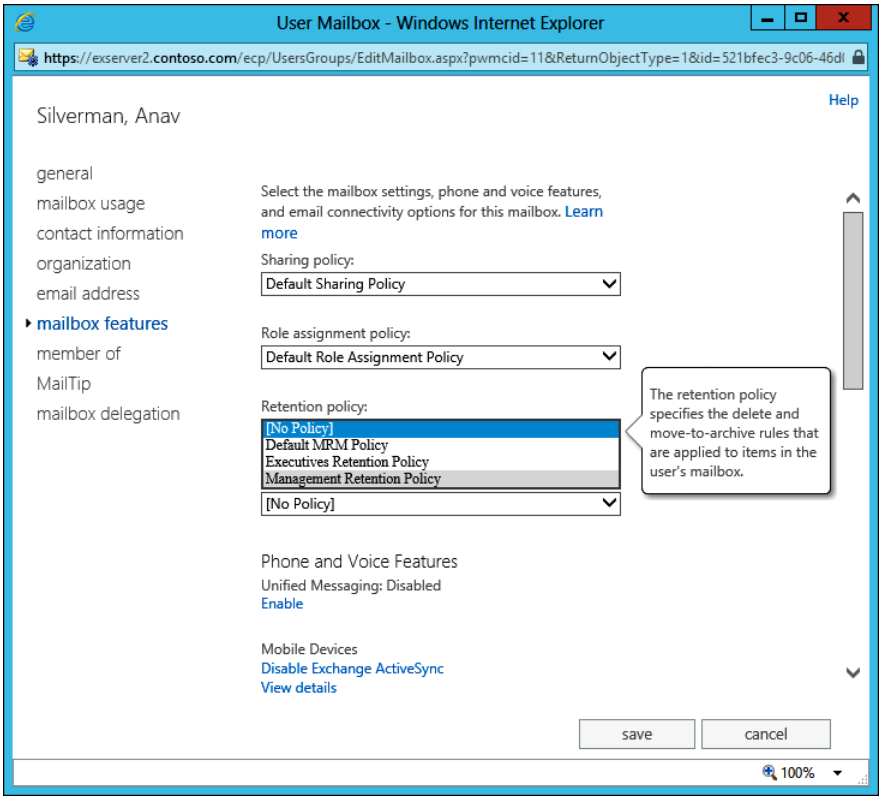


Figure 11-13 Assigning a retention policy to a mailbox

INSIDE OUT Only one retention policy—ever

A mailbox can only ever have one retention policy, so when you assign a retention policy to a mailbox, the action overwrites any policy that might already be in place. You can change retention policies multiple times on a mailbox, but this isn't a good idea unless you really need to switch policies because the effect of the different policies might confuse users; the MFA responds to different retention settings in the different policies. Setting a value for the RetentionURL parameter is not compulsory, but it is a useful way to communicate how a user might find additional details about the company's retention policy. This URL is visible only through Outlook 2010 and Outlook 2013 (Figure 11-14) and isn't displayed by earlier clients or Outlook Web App.

```
Set-Mailbox –Identity 'TRedmond' –RetentionPolicy 'Management retention policy'
–RetentionComment 'Management retention policy applies to this mailbox' –RetentionURL
'http://Intranet.contoso.com/RetentionPolicies.html'
```

By looking at the parameters that set retention information with the Set-Mailbox command, you can see that the RetentionComment property provides the text you can see beside Account Settings. The RetentionUrl property populates the URL for the More Information link, hopefully taking the user to a website on which she can find some additional information to explain why a retention policy is necessary and what it means to the user. It is good practice to set up such a website and populate it with some practical examples of what a retention policy means to a user. For instance, it could explain to users that items in some of the default folders are automatically cleared to the archive after a certain period. Taking such a step might just save some expensive help desk calls!

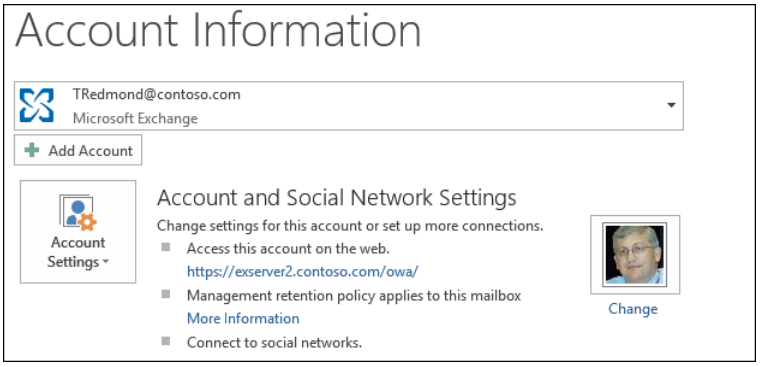


Figure 11-14 How Outlook shows retention policy information in its backstage area

It is common practice to set these properties when you place a mailbox on any type of hold (in-place or retention). These topics will be discussed in the following sections. For now, although you don't have to set these properties to impose an effective retention regime, they are helpful to communicate information to users about what's going on in their mailbox. Experience with many projects demonstrates that anything that assists in effective communications with users is likely to reduce help desk calls.

The value of $Null

When you want to remove a retention policy from a mailbox, you just set its policy to *$Null*. For completeness, it's a good idea to also set the other properties associated with retention policies to $Null. Here's the command:

```
Set-Mailbox –Identity 'JSmith' –RetentionPolicy $Null –RetentionComment $Null
–RetentionURL $Null
```

After you begin to deploy retention policies to mailboxes, the question arises of how to integrate the assignment of retention policies with any user provisioning process your company has in place. Unless you also enable an archive for a mailbox when it is created, Exchange won't assign a default retention policy automatically, so an explicit administrative action is usually required to allocate a retention policy to a mailbox. This action is not difficult to code with EMS, but it is something that needs to be considered as part of your deployment plan.

INSIDE OUT **Managing retention policies and tags in a hybrid environment**

If you run a hybrid environment in which some mailboxes are on-premises and some are on Office 365, you have to synchronize retention policies and tags from the on-premises organization to the cloud tenant domain to ensure that the same policy applies to all customers. You can create the retention policies and tags within Office 365 or you can use the scripts provided by Microsoft. Export-RetentionTags.ps1 exports details of retention policies and tags from an on-premises organization to a file in XML format, and Import-RetentionTags.ps1 imports the data from a previously exported file. Regretfully synchronization is not automatic and has to be done on a regular basis.

Modifying a retention policy

Policies can evolve over time by the addition or removal of tags. The easiest way to modify a retention policy is to edit it with EAC; the console will take care of all complexities involved in selecting whatever tags you want to add to or remove from the policy and then making sure that the tags are processed correctly. If you want, you can do the same work with EMS, but this is not recommended because great care must be taken to ensure that the retention tags are written correctly into the policy.

Customizing retention policies for specific mailboxes

You can tailor the retention policy for a specific user by assigning personal tags on a per-mailbox basis. You can do this only if a retention policy already applies to the user's mailbox. For example, assume that you want to assign a new personal tag to a user so he can mark an item to be moved into the archive after a year. You can do this as follows:

```
Set-RetentionPolicyTag –Mailbox JSmith –OptionalInMailbox '5 Year Delete'
```

Exchange adds the optional tag to the set of tags covered in the retention policy that already applies to the mailbox and makes the expanded set available the next time the user connects. Unfortunately, no cmdlet is available to report whether a mailbox has been assigned optional tags. If you examine a mailbox with Get-Mailbox, it tells you only whether a retention policy is assigned. Therefore, if you want to change the list of optional tags assigned to a mailbox, you have to write the complete list with Set-RetentionPolicyTag. For example, to add a tag to the one that has already been assigned, use this command:

```
Set-RetentionPolicyTag –Mailbox JSmith –OptionalInMailbox '5 Year Delete', '2 Year
Delete'
```

EMS doesn't validate that the tags you assign to a mailbox will be effective. For example, you can assign a new archive tag to a mailbox that doesn't have a personal archive. This is really a null operation because neither Outlook Web App nor Outlook displays archive tags if the mailbox doesn't have a personal archive.

To remove all optional retention tags from a mailbox, set the list to *$Null* as follows:

```
Set-RetentionPolicyTag –Mailbox JSmith –OptionalInMailbox $Null
```

INSIDE OUT Accessing personal tags through Outlook Web App

Exchange makes accessing personal tags easier by allowing users to see a list of available personal tags through the Outlook Web App Options list to decide which personal tags they would like to use. Figure 11-15 shows the option exposed through the Organize Email section of Outlook Web App Options. A retention policy must be in effect for a user's mailbox, and the MyRetentionPolicies setting of the user role assignment policy (RBAC) that applies to the mailbox must be allowed (Figure 11-16) before Outlook Web App reveals personal tags. If shown, the user sees the personal tags that she can already use because they are included in the retention policy (listed as Required) and the other personal tags that are defined for the organization that she can choose to use (listed as Optional). The user cannot remove any of the Required tags because their presence is mandated by the retention policy that is applied to the mailbox. A user can begin to apply personal retention tags to items immediately after adding the tags to her mailbox.

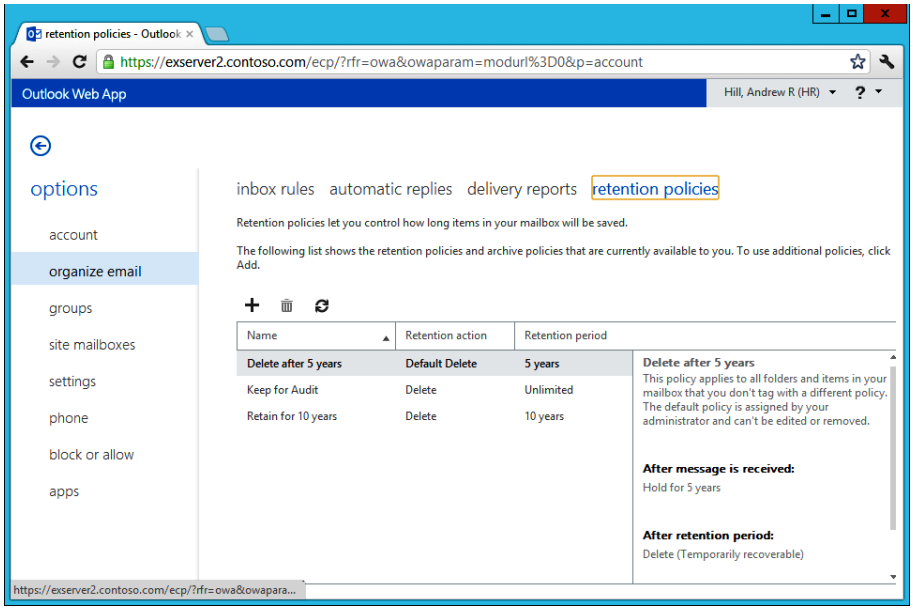


Figure 11-15 User access to personal retention tags through Outlook Web App Options

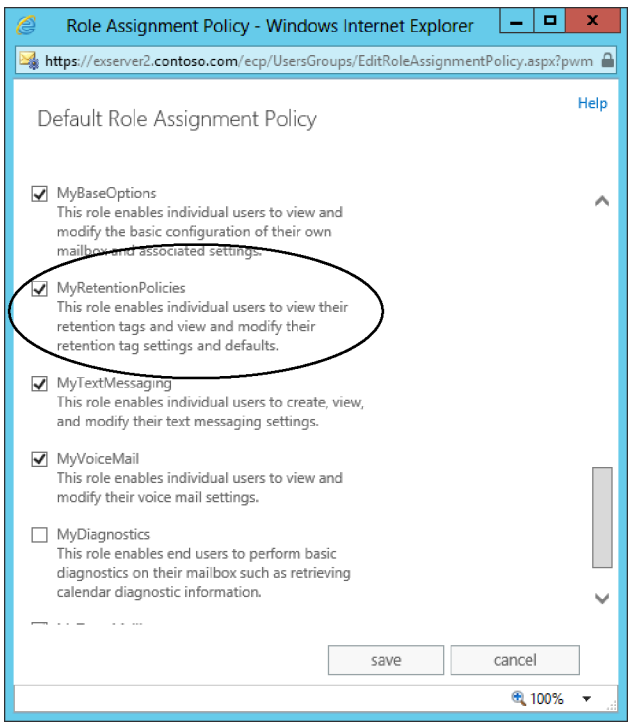


Figure 11-16 Amending a user role assignment policy to reveal personal retention tags

User interaction with retention policies

The first evidence users see that their mailbox has been assigned a retention policy is when retention information appears when they look at messages. This information is based on the tag stamped on an item by the MFA. Thirty days or so before an item's retention period expires, Outlook and Outlook Web App begin to inform users that they might want to take action to preserve the item; otherwise, the MFA will process it again and delete it or move it into the archive, depending on the action required by policy. These warnings are visible when a message is opened or shown in the message preview. Figure 11-17 shows how Outlook Web App advises that a message has five days before it expires as the result of the retention policy tag placed on the Inbox. The user now has the choice either to take action or to let the message expire, in which case the MFA will process whatever action is defined in the tag.

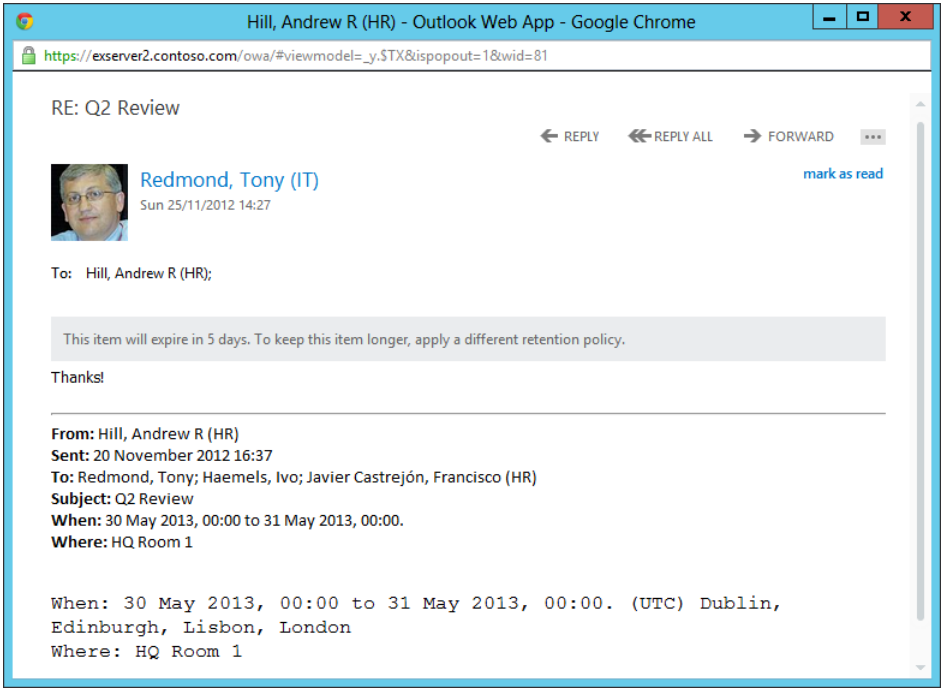


Figure 11-17 Outlook Web App warning that an item is approaching its expiry deadline

Users have two options. First, they can move the item to a different folder and so remove it from the influence of the retention policy tag that applies to Inbox items. After it is moved, the item is governed by the default policy tag defined in the retention policy that applies to the mailbox, if one exists, or by an explicit tag that is applied to the folder and therefore inherited by all items that are added to the folder. If neither of these conditions exists, the item is left untagged and is therefore not subject to processing by the MFA.

The second option is to apply a personal tag to the item. Users can choose from any of the personal tags defined in the retention policy applied to their mailbox by right-clicking an item and then selecting the personal tag to apply.

You won't see the user interface for retention policies unless a policy is applied to your mailbox and the MFA has processed the mailbox to apply the retention policy. As part of this process, the MFA creates a hidden folder-associated item (FAI) in the user mailbox that clients use to populate the retention tag picker. If the policy is subsequently updated with a new retention or archive tag, the new tag will not be visible to clients until after the MFA next processes the mailbox.

After a personal tag has been applied to an item, the item is no longer subject to the provisions of either the folder policy or the default policy because an explicit tag always takes precedence over a tag placed on a folder. The personal tag also remains with the item if it is moved to another folder or into the personal archive. If users want to impose a different retention policy on the item, they must replace the existing tag with a new personal tag.

The Managed Folder Assistant automatically applies the retention policy

Although most retention policies include a default policy tag to provide a catch-all retention action for items not tagged by other means, Outlook and Outlook Web App support the use of personal retention tags to set a specific retention policy on a folder. Exchange applies the policy defined in the personal retention tag to items held in the folder in much the same way it applies the retention policy tags placed on default folders such as the Inbox. In some respects, you can use this approach to create a roughly equivalent situation to the functionality that Exchange 2007 managed folders provides. However, you have to create the folders and apply the retention policies manually, whereas the MFA does the work to push out new folders to user mailboxes and apply the retention policy automatically for managed folders.

Setting a retention policy on a folder

Default folders such as the Inbox or Sent Items probably come under the control of a folder retention tag included in the retention policy that's applied to a mailbox. You cannot override the policy set on default folders, but you can apply a different policy to user-created folders by assigning a personal tag to the folder. All the items that are held in the folder will then inherit the tag placed on the folder unless they in turn are tagged with a different personal tag.

To set a new policy for a folder by applying a personal tag using Outlook 2013, select the folder and click the Assign Policy icon on the toolbar. Outlook then displays a list of the personal tags. Click one of the tags to select it. Outlook puts a check mark beside the chosen tag to indicate that it will apply this tag to the folder; the items in the folder will inherit the tag from the folder. Another way of accomplishing the task is to select the properties of a folder by right-clicking the folder in the Outlook folder list and then choosing to the Policy tab (Figure 11-18).

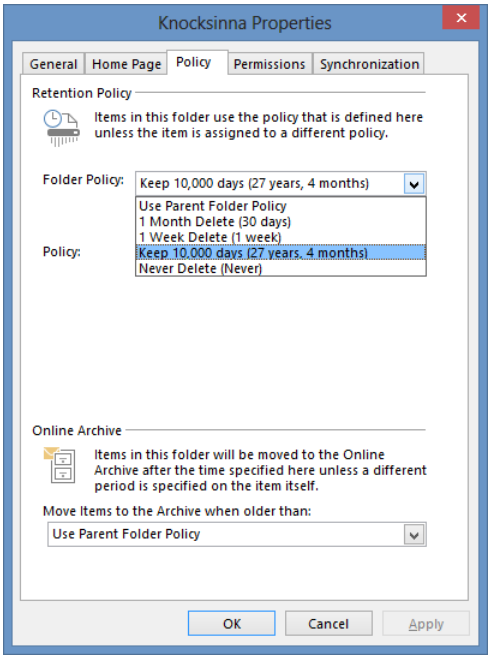


Figure 11-18 Applying a personal retention tag through folder properties

If a policy tag is in force for the folder, it will be shown as the current folder policy. If you attempt to apply a personal tag on a default folder that is controlled by a retention policy, you won't be able to select a new personal tag to apply. If the folder policy is listed as Use Parent Folder Policy, no specific personal tag has been applied to the folder, and the MFA will apply the default retention tag to the items in the folder. This is also the case for archive policies, shown in the Online Archive area at the bottom of the screen, where you can see that the folder uses the default archive tag.

You can select any personal tag available to the mailbox through its assigned retention policy and apply it to the folder. Outlook displays the available personal tags in a drop-down list (Figure 11-18). When selected, Outlook updates the retention data stored in the mailbox (which makes the new policy visible to the user). In the background, the MFA applies

the policy to the folder and the items contained in the folder the next time it processes the mailbox unless an explicit personal tag has already been applied to an item in the folder, in which case the existing tag is retained. To set a default policy on a folder with Outlook Web App, select the folder from the folder list under the Mailbox root, right-click to select Apply Policy, and then select the personal tag to apply to the folder.

Removing tags from policies

You remove a retention tag from a policy by selecting the policy in EAC and editing the list of tags, removing the one you no longer use. When you remove a tag like this, the MFA erases details of the tag from the policy information that is held in user mailboxes. This action makes the tag unavailable (invisible) to the user. If the tag is replaced by another tag, like when the tag for a default folder is replaced by another tag for the same folder, MFA applies the new tag to the folder the next time it processes the mailbox.

However, if a personal tag is removed from a policy, the MFA leaves the tag in place on whatever items to which the tag has been applied and continues to process the items according to the policy expressed with the now-removed tag. This is because the tag is still a known object within the Exchange configuration data, so the tag stays in effect until it is replaced in some way, such as by a different personal tag being applied. Despite the fact that the user can no longer apply this personal tag to other items, it will be shown to the user if he views an item to which the tag was applied when it was included in the policy.

If you delete a tag, it is removed from Active Directory. The MFA cannot process a nonexistent tag, so it has to search all mailboxes to discover items that were stamped with the now-deleted tag and restamp these items with whatever tag is now applicable, such as the default tag. Searching a Mailbox server to locate and then process potentially hundreds of thousands of items is obviously not good for server performance, and this will happen across every Mailbox server in the organization unless the deleted tag was used only for mailboxes on specific servers, which is very hard to predict and control. Therefore, you should not delete a tag without good reason. If you want to remove a tag, it is always better to just remove the tag from the policies when it is present rather than deleting it from the organization.

You can also disable a tag by editing its properties to set the retention period to Never. The MFA still considers the tag to be valid, but it ignores items that are stamped with the tag and never deletes or moves these items. In addition, although the tag is disabled, the MFA will not apply the default tag in the policy to these items, so they are essentially in limbo with respect to retention. Disabling a tag is a good way of stopping the MFA from processing items while keeping the items stamped so that they can be reactivated for retention purposes by just editing the tag to set its retention period to some number of days.

Changing a retention tag: An exception to the rule

Changing the retention period in a tag is similar to removing a tag. All items stamped with the retention period up to when you made the change continue to use that retention period; items that are stamped with the updated policy will have the new retention period. The exception to this rule is when you set the RetentionEnabled property of a tag to $False; this value instructs the MFA to leave the tags in place but ignore them when it processes items. For example:

```
Set-RetentionPolicyTag –Identity 'Keep Items Forever' –RetentionEnabled $False
```

This situation continues until the user explicitly assigns a replacement tag to an item or you remove the tag from Active Directory by using the Remove-RetentionPolicyTag cmdlet. When this happens, the MFA removes the deleted tag from any items on which it was used the next time it runs.

Removing a retention policy

The Remove-RetentionPolicy cmdlet removes a retention policy from the organization. For example:

```
Remove-RetentionPolicy –Identity 'Retention Policy - PR Department'
```

Removing a retention policy removes the policy from any mailboxes to which it is currently applied. If any mailboxes are associated with the policy, EMS prompts you to confirm its removal. If you proceed, Exchange removes the reference to the now-deleted policy from the mailboxes. Exchange can't decide what retention should replace the one that has just been removed, so no policy is applied. Locating the mailboxes to which a retention policy is applied is therefore a proactive step you should take before you remove the policy. You can scan mailboxes to discover where a retention policy is applied with a command such as this:

```
Get-Mailbox | Where {$_.RetentionPolicy –eq "Retention Policy - Audit Department"} |
Select Name
```

A similar set of commands can be run to locate mailboxes with a specific retention policy and assign a new retention policy to the mailboxes. For example:

```
Get-Mailbox | Where {$_.RetentionPolicy –eq "Retention Policy – Audit Department"} |
Set-Mailbox –RetentionPolicy 'New Retention Policy for Auditors'
```

Upgrading from managed folders

Managed folders are now a deprecated feature, and the code that supports these folders will be removed from Exchange in a future version. It's therefore important to upgrade managed folders to equivalent retention tags as soon as possible. You can do this by using

managed folders as templates to create new tags. For example, assume that you have a managed folder called Never Delete that acts as a repository for items that users never want to have removed from a mailbox because they are so important. You could argue that these items could be stored in an archive mailbox. However, archive mailboxes didn't exist in Exchange 2007, and it takes time for people to change their behavior. You can use a command such as the one shown here to create a new retention policy tag from the Never Delete managed folder:

```
New-RetentionPolicyTag –Name 'Mark item to never expire' –ManagedFolderToUpgrade
'Never Delete' –Comment 'Tag created from old Never Delete managed folder'
```

To complete the process, you must associate the new tag with a retention policy and assign it to a user, and then the user can apply the new tag to any item in his mailbox rather than just to the items placed in the managed folder.

How the Managed Folder Assistant implements retention policies

After you apply a retention policy to a mailbox, you can either wait for the next scheduled run of the MFA or start it manually so that the new policy is applied immediately. When the MFA runs, it performs the following tasks:

- It applies the tags specified in retention policies to the mailboxes covered by these policies and stamps the items in the various folders covered by the policies with the appropriate tag name and expiration date.
- If a policy defines a retention or expiry period for items, it stamps a Messaging Application Programming Interface (MAPI) property (ElcMoveDate) on the items, indicating the date and time from which the retention period will start. A future run of the MFA can then use this date and time to calculate when to delete an item or mark it as expired.
- It locates items in folders that are past their expiration date and takes whatever action is defined in the policy (delete, age out, move to another folder).

The MFA works on a work cycle basis: it assesses the expected workload in terms of the number of mailboxes it has to process and then spreads out its processing across the complete window. For example, if 600 mailboxes are to be processed over three hours, the MFA creates its own internal schedule to process 200 mailboxes per hour or roughly three mailboxes per minute. In addition, a checkpoint is defined for the work cycle, at which time the MFA looks for new mailboxes that should be added to its list for processing. The default values for the work cycle and checkpoint are both 1 day, so the MFA attempts to process every mailbox in its list daily and checks for new mailboxes daily. The MFA logs event 9017

when it begins to process mailboxes, but the more important event to look for is 9025, logged when the MFA has had to skip a mailbox for some reason, such as when the mailbox is being moved to another database.

Overall, the work cycle mechanism makes more effective use of server resources in an easy and relaxed manner throughout the day and doesn't create potential spikes in demand.

You might want to run the MFA immediately, perhaps to apply a policy to a group of users for the first time. Forcing immediate execution for a selected mailbox is useful when you start to apply policies to mailboxes and want to gauge the effect of the policy by examining the contents of a known mailbox. This might be easier than asking users what happened to items in their mailboxes (especially if you've made a mistake with the policy and just removed half the items from the mailbox). To force processing for a selected mailbox, specify its name with the –Identity parameter:

```
Start-ManagedFolderAssistant –Identity 'Akers, Kim'
```

To process a group of mailboxes, either provide a set of mailbox identifiers as input or use the Get-Mailbox cmdlet with a filter to retrieve a set of mailboxes and pipe it as input to Start-ManagedFolderAssistant. In the first example, two mailbox identifiers are provided as input. In the second, you process all the mailboxes in a database. In the third, you use a filter to find all the mailboxes from a particular office.

```
"Redmond, Tony", "Akers, Kim" | Start-ManagedFolderAssistant
Get-Mailbox –Database 'VIP Data' | Start-ManagedFolderAssistant
Get-Mailbox –Filter {Office –eq 'Dublin'} | Start-ManagedFolderAssistant
```

The time required for the MFA to complete its run depends on the number of mailboxes and the number of items to which it has to apply retention policies. A run on a small server that hosts a few hundred mailboxes will complete in a couple of minutes unless the mailboxes hold thousands of items. However, processing 7,000 mailboxes, each of which holds an average of 20,000 items, could take several hours, especially if the server is loaded with other tasks or the policies cause a heavy I/O load because many items are permanently removed or moved from primary to archive mailboxes. You should monitor the first runs of the MFA on a server to gauge the scope of the activity and how long a normal run takes to complete. Equipped with this information, you can quickly assess whether future runs are progressing as expected.

After the MFA has applied a new policy to a mailbox, the next time the user connects to the mailbox with a client that supports retention policies, she will see that retention tags are shown on items and the retention policy options are visible. Another important point that you should understand is that if you apply a retention policy that contains a default policy tag, the MFA stamps the default tag on every item in the mailbox. This action forces Outlook to download the complete contents of the mailbox the next time the client

connects and synchronizes with Exchange. Such a massive synchronization has the potential to flood a network and keep clients fully occupied for a long time. Including a default archive tag in a policy does not have the same effect because the MFA does not stamp every item with this tag.

Behind the scenes with the MFA

When the MFA first processes a mailbox, it creates a hidden FAI item of type IPM.Configuration.MRM in the Associated Contents table of the Inbox folder. The item stores MRM configuration in XML format in a PR_ROAMING_XMLSTREAM property. (PR = property; Roaming means that the information is available wherever the client connects, and XMLSTREAM indicates the type.) The MFA then updates the item any time a change occurs in the retention policy assigned to the mailbox, such as when a new personal tag is added to the policy. If a new retention policy is assigned, the MFA updates the item with details of that policy.

Exchange uses FAIs as a means to hold configuration and other data it needs to store in a mailbox but does not want to reveal to users when they run clients such as Outlook. The item the MFA creates holds details of the retention policy that has been assigned to the mailbox, including details of the retention tags the client needs to display in its user interface. If the item does not exist in the mailbox, a client remains unaware that a retention policy is in force. This is why the MFA has to process a mailbox before the client user interface is populated with details of the policy. In addition to the tags provisioned through the retention policy, the item also holds information about any personal tags the user has selected for his own use through Outlook Web App Options (see Figure 11-15). As noted earlier, you cannot see this information with a normal client, but you can with the MFCMAPI utility by opening a mailbox that has been assigned a retention policy, opening the Inbox folder, and then opening the associated contents table before finally identifying the MRM item in the set held in the table. Figure 11-19 shows the details of the MRM configuration item as exposed by MFCMAPI, and you can clearly see details of the retention tags specified in the policy in the text box.

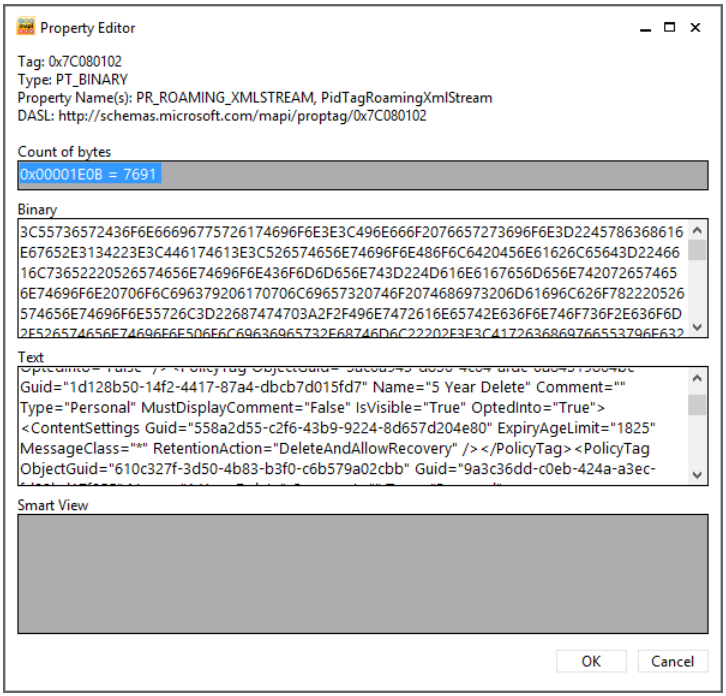


Figure 11-19 Viewing the MRM configuration item with MFCMAPI

The MFA uses a number of MAPI properties for mailbox folders and items during its processing. These properties are:

- **PR_START_DATE_ETC** For a folder, this property holds the GUID of the retention tag that governs a folder. Every tag is assigned a GUID (stored logically in the GUID property and visible with EMS). This GUID is stamped into this property so the MFA knows which retention tag, action, and period applies to the folder. For an item, the property holds a composite value containing the default retention period plus the start time for the retention period. By adding the retention period to the start time, the MFA determines the expiry date.

- **PR_RETENTION_PERIOD** This is the number of days items should be retained in a folder. When a personal tag is applied to an item, this property is set. However, if the property does not exist, the item (or subfolder) inherits the retention period from its parent folder.

- **PR_RETENTION_DATE** This is the calculated date when an item's retention period expires. Clients display this information to users. When clients work in online mode (such as in Outlook Web App), Exchange takes care of calculating this value, otherwise clients that work in offline mode perform the calculation. otherwise.

- **PR_RETENTION_FLAGS** For a folder, this flag indicates whether the retention tag is inherited from the parent folder. If the user sets an explicit tag on a folder, the value is nonzero.
- **PR_POLICY_TAG** This exists only for items and contains a binary encoded value pointing to the policy tag that governs the item.
- **PR_ARCHIVE_TAG** This exists only for items and points to the archive tag that governs an item.
- **PR_ARCHIVE_PERIOD** This exists only when an item has been stamped with an explicit archive tag to contain the number of days in the archive retention period.
- **PR_ARCHIVE_DATE** This contains the date when an item will be archived.

You can view these properties through MFCMAPI. Figure 11-20 shows the value of the PR_RETENTION_DATE property for an item.

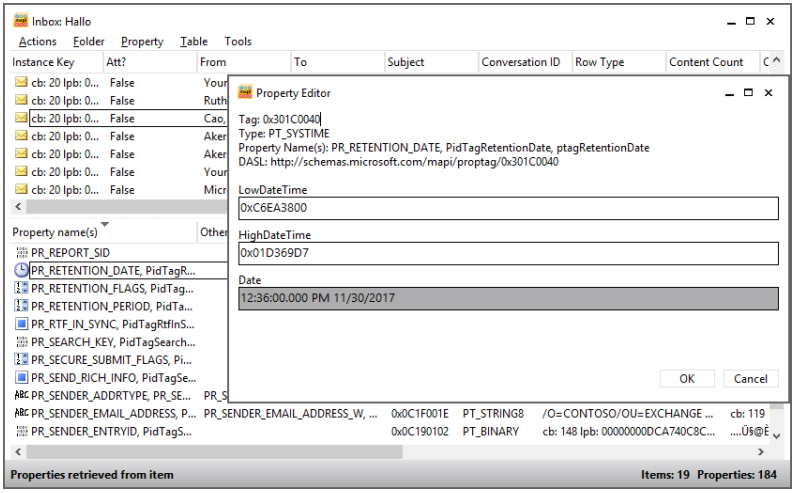


Figure 11-20 Viewing the PR_RETENTION_DATE property for an item

Retention date calculation

Date calculation is a very important part of the work the MFA does when it processes items. The MFA has to understand the date that should be used to calculate the age of the item (when the item first appears in the mailbox) and either the date when the item will expire or the date when the MFA has to take the action the retention policy requires (deleting the item or moving it into the archive).

Given the nature of email, most items enter a mailbox when they are delivered in the Inbox or are sent and stored in Sent Items. Items often stay in these folders for much of their lifetime and, if they do stay, will probably come under the control of either a folder tag (if defined for the folder) or a default tag (that applies to untagged items in the mailbox). The exception, of course, occurs when a user explicitly applies a personal tag to an item in the Inbox or Sent Items.

When the MFA runs, it examines items in the mailbox and determines what processing is required. Assume that an Inbox item exists that was delivered on 1 April 2013 and that a folder tag that requires items to be deleted and allow recovery after 30 days is applied to the Inbox. MFA stamps this item with a start date of 1 April 2013 by updating the PR_START_DATE_ETC property. It then calculates the expiration date by adding 30 days to the start date; 1 May 2013 is the result. This date is then stamped on the item by updating the PR_RETENTION_DATE property. On or after 1 May 2013, the MFA returns to the item and discovers that its retention period has expired. (If you view the item by using Outlook before the MFA returns, you'll see the item marked as Expired.) When the MFA processes an expired item, it takes the retention action defined in the tag. In this case, it moves the item into the Recoverable Items folder, where it will remain until the retention period for the mailbox expires, at which time the MFA will remove the item permanently from the database.

This is the simplest kind of processing the MFA is called on to perform, but it happens for a large percentage of items because many users leave messages in the Inbox and Sent Items folders. People who let items accumulate in these folders are often called pilers because they create piles of messages and then rely on client search facilities to locate specific items when required.

Broadly speaking, the other kind of user behavior is represented by the filers, or people who move items from the Inbox and Sent Items into more appropriate folders in which the items form collections that represent important categories of work (or play) as viewed by the user. Or indeed the user attempts to keep the Inbox and other folders under some form of control by deleting unwanted items on a regular basis.

Assume that your user receives a message, reads it, and then comes back to the message a week or so later when its content is no longer required, perhaps because he has acted on a request that was contained in the message. Because the item has been in the mailbox for more than a day, the MFA should have already processed it and stamped its properties with a start and an end date for retention purposes. The user now deletes the message as part of his cleanup effort, and the client moves the item into the Deleted Items folder.

You now have a different retention context because the item has moved out of the sphere of control exerted by the folder tag placed on the Inbox. Retention policies usually include a folder tag to control the Deleted Items folder and, if so, the MFA now must apply the

instructions contained in that tag. Otherwise, the MFA will apply the instructions contained in the default tag.

Assume that the folder tag assigned to Deleted Items requires items to be removed from this folder (using Delete And Allow Recovery) after seven days. The MFA examines the item and discovers that its start date is 1 April 2013 and that it should now be removed on 8 April 2013. If the user moved the item into Deleted Items before 8 April 2013, the item is kept until 8 April 2013 and then removed. However, if the MFA processes the items on 8 April 2013 or after, it knows that the expiration date for the item is already past and therefore immediately executes the retention action and moves the item into the Recoverable Items folder, where it will remain until the deleted items retention period specified for the database elapses. The default deleted items retention period for newly created databases on an Exchange 2013 server is 60 days. However, this period is calculated based on the item's deletion date rather than its creation date, so it will be held for another 60 days to allow the user to recover the item during this period. (For example, if an item is deleted on 2 April 2013, the MFA will purge it on 1 June 2013.) Some companies use an extended deleted items retention period of up to a year to ensure that users can recover deleted items themselves without involving administrators, a useful practice that costs a little more disk space for databases while releasing administrators from the tedium of having to restore databases to recover a now-important item deleted some months in the past.

Calendar and task items pose some particular difficulties for the MFA when it comes to calculating their expiration date, which is why Microsoft took some time to find the best way for the MFA to proceed and implemented only the processing from Exchange 2010 SP2 RU4. Calendar items are either nonrecurring or recurring and will be either in the Calendar folder or in Deleted Items. It is possible that calendar items might turn up in other folders, but this happens very rarely in practice; if it does, the MFA deals with these items as if they are in the Calendar folder. The following rules are observed:

- Nonrecurring items in the Calendar folder expire according to the end date of the appointment or meeting. For example, if you create a calendar entry for a trip that lasts from 1 June 2013 to 10 June 2013 and the folder tag for the Calendar folder specifies a retention period of two years, the item will expire on 10 June 2015.
- Recurring items in the Calendar folder expire according to the end date of their last occurrence. For example, if a series of meetings is scheduled from 1 May 2013 to 1 September 2013, the item expires on 1 September 2015. If a recurring event has no end date, it will never expire.
- Calendar items in the Deleted Items folder expire based on their message-received date. If this property is not populated (for instance, for an item the user created in his own calendar and never sent to anyone else), the MFA bases its decision on the message-creation date. If neither of these dates is populated, the MFA ignores the item.

Task items receive similar care when the MFA calculates their expiration. The following rules are used:

- Nonrecurring (simple one-off) tasks stored in any folder other than Deleted Items expire based on the message-received date. If this property does not exist, the MFA uses the message-creation date.
- Recurring tasks expire based on the date of their last occurrence. A recurring task with no end date will never expire.
- A regenerating task (a special form of recurring task that generates a specified time after the preceding instance of the task is completed; see Figure 11-21) never expires. Only the most dedicated of users who understand how to make maximum use of tasks create these items!
- Tasks that have been placed in the Deleted Items folder expire based on their message-received date. If this property does not exist, the MFA uses the message-creation date. If neither date is available, the task never expires.

As explained in the previous section, you can use MFCMAPI to validate the dates the MFA uses to calculate item expiration.

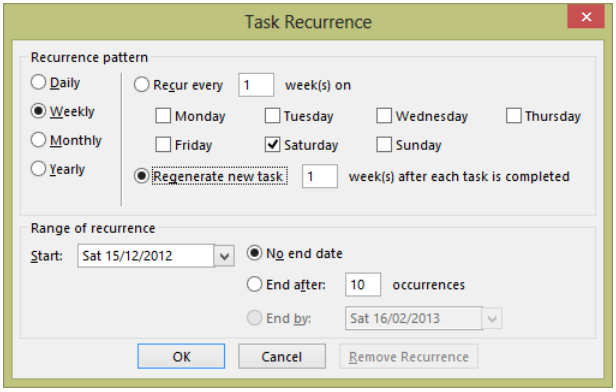


Figure 11-21 Specifying a regenerating task

Preserving information

An important part of being able to achieve a compliance strategy is the ability to preserve information. Retention policies help somewhat because they can automatically move information into archive mailboxes. However, they also remove information through tags that require items to be moved into the Recoverable Items folder or permanently deleted. Items