Exam Manual

LearnSmart

Microsoft Exchange Server (70-662) 2010 Configuration

This LearnSmart exam manual breaks down complex topics and provides candidates with all the knowledge and confidence required to successfully complete the Exchange Server 2010 Configuration exam (70-662). By studying this guide, candidates will become familiar with the exam's format and strengthen their skill sets with respect to an array of topics, including:

- Installing and Configuring Exchange Server
- Configuring Exchange Recipients and Public Folders
- Configuring Client Access
- And more!

Give yourself the competitive edge necessary to further your career as an IT professional and purchase this exam manual today!

Exchange Server 2010 Configuration (70-662) LearnSmart Exam Manual

Product ID: 012467
Production Date: July 13, 2011

Warning and Disclaimer

Every effort has been made to make this document as complete and as accurate as possible, but no warranty or fitness is implied. The publisher and authors assume no responsibility for errors or omissions. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this document.

Volume, Corporate, and Educational Sales

Favorable discounts are offered on all products when ordered in quantity. For more information, please contact us directly:

1-800-418-6789
solutions@learnsmartsystems.com

International Contact Information

International: +1 (813) 769-0920

United Kingdom: (0) 20 8816 8036

Table of Contents

Abstract

LearnSmart's Exchange Server 2010 Configuration (70-662) Exam Manual is intended to help complement and augment the reader's training program for Microsoft certification exam 70-662 TS: Microsoft Exchange Server 2010, Configuring. Being consistent with the audience profile for the exam, this guid assumes that the reader is already familiar with Exchange 2010 with a similar lever of familiarity as someone who " is responsible for the maintenance and administration of the Exchange servers in an enterprise environment." This Exam Manual focuses primarily on the seven domains TS: Microsoft Exchange Server 2010, Configuring (70-662) exam.

What to Know

The TS: Microsoft Exchange Server 2010, Configuring (70-662) exam covers nearly all aspects of configuring Exchange Server 2010. The only area that is conspicuously lacking is Unified Messaging. Exam participants and individuals in Microsoft Exchange job roles are typically expected to know what Unified Messaging is, but this exam does not test on Unified Messaging configuration.

Tips

There are some excellent training resources available for this exam, but stick to using reputable sources. The Internet is filled with inaccurate information about this exam, and about the material that it covers.

Domain 1: Installing and Configuring Exchange Server

Prerequisite Planning

Before you can install Exchange Server 2010, there are several prerequisites that must be in place. The prerequisites include:

The server on which Exchange Server 2010 will be installed must be running a 64-bit edition of Windows Server 2008 (SP2 or higher) or Windows Server 2008 R2. A full installation of Windows Server is required, as Exchange 2010 cannot be installed on top of a Server Core installation.

If you are going to be deploying a Mailbox Server, and that server will be a part of a Database Availability Group (DAG), then the server must be running either Windows Server 2008 Enterprise Edition (SP2 or higher) or Windows Server 2008 R2 Enterprise Edition. The Standard Edition of Windows Server cannot be used for mailbox servers that will be a part of a Database Availability Group (DAG).

It is important to plan your Exchange Server deployment carefully, because Windows Server cannot be upgraded after Exchange has been installed.

There are several prerequisite components which must be installed on your Windows Server prior to installing Exchange. These prerequisites differ depending on the version of Windows that you are using.

Windows Server 2008 Prerequisites

The prerequisite components for Exchange Server deployments running on Windows Server 2008 include:

- Microsoft .NET Framework 3.5 SP1
- The Microsoft .NET Framework 3.5 Family Update for Windows Vista x64 and Windows Server 2008 x64
- Windows Remote Management (WinRM) 2.0
- Windows PowerShell 2.0

If the server will be running the Hub Transport or the Mailbox Server roles, then the Microsoft Filter Pack is required.

Preparing the Operating System

Besides the prerequisites that have already been listed, there are a number of operating system components which must be installed prior to installing Exchange. Microsoft provides a number of scripts that are designed to automate the process of deploying the necessary operating system components. The scripts are located in the Scripts folder on the Exchange 2010 installation DVD. To run the scripts, you must open an elevated Command Prompt window and then navigate to the DVD's Scripts folder. The scripts that you will run vary depending on the Exchange Server roles that you plan to deploy. Although the required operating system components can be deployed manually through the GUI, you should expect some test questions regarding script based preparation of the operating system.

Here is a summary of the role configurations and their corresponding scripts:

If your Exchange 2010 Server will host a typical installation (Hub Transport, Client Access, and Mailbox Server roles) then you would prepare the server by using these commands:

```
sc config NetTcpPortSharing start= auto
ServerManagerCmd -ip Exchange-Typical.xml -Restart
```

If you are planning on running the Hub Transport, Client Access, Mailbox, and Unified Messaging Server roles, then you must run these commands:

```
sc config NetTcpPortSharing start= auto
ServerManagerCmd -i Desktop-Experience
ServerManagerCmd -ip Exchange-Typical.xml -Restart
```

If your server will be configured with the Hub Transport and Client Access Server roles only, then this is the command that you would use:

```
sc config NetTcpPortSharing start= auto
ServerManagerCmd -ip Exchange-Typical.xml -Restart
```

If the server will only host the Hub Transport and Mailbox Server roles, then you would use the following command:

```
ServerManagerCmd -ip Exchange-Typical.xml -Restart
```

Servers hosting the Client Access and Mailbox Server roles require the following commands:

```
sc config NetTcpPortSharing start= auto
ServerManagerCmd -ip Exchange-Typical.xml -Restart
```

You would run the following commands to prepare dedicated Client Access Servers:

```
sc config NetTcpPortSharing start= auto
ServerManagerCmd -ip Exchange-CAS.xml –Restart
```

The following command is used for preparing a dedicated Hub Transport Server:

```
ServerManagerCmd -ip Exchange-Hub.xml -Restart
```

If the server will act as a dedicated Mailbox Server, then you should use this command:

```
ServerManagerCmd -ip Exchange-MBX.xml -Restart
```

Dedicated Unified Messaging Servers can be prepared by using this command:

```
ServerManagerCmd -ip Exchange-UM.xml -Restart
```

Edge Transport Servers can be prepared by using the following command:

```
ServerManagerCmd -ip Exchange-Edge.xml -Restart
```

Windows Server 2008 R2

Although there is a long list of prerequisite components that must be deployed prior to installing Exchange Server on Windows Server 2008 SP2, the list is much shorter for Windows Server 2008 R2 servers. The only required external component is the Microsoft Filter Pack. Furthermore, the Microsoft Filter Pack is only required for servers that will host the Hub Transport or Mailbox Server roles.

Preparing the Operating System

As was the case with Windows Server 2008 SP2, there are some operating system components that must be in place before you can begin installing Exchange. As is the case with Windows Server 2008 SP2, you can deploy these components by using a scripted installation. However, the required scripts and the method that you will use to execute them are different.

Before you can begin deploying the required operating system components, you must open an elevated PowerShell window and import the Server Manager module by using this command:

```
Import-Module ServerManager
```

Different Exchange Server roles require different Windows components to be installed. Therefore, the commands that you will use vary depending on the Exchange Server roles that the server will be hosting.

If the server will be running a typical configuration (Hub Transport, Client Access, and Mailbox Server roles) then you must use these commands:

```
Add-WindowsFeature NET Framework      WAS-Process-Model
RSAT-ADDS                             RSAT-Web-Server
Web-Server                            Web-ISAPI-Ext
Web-Basic-Auth,Web-Windows-Auth       Web-Digest-Auth
Web-Metabase                          Web-Dyn-Compression
Web-Net-Ext                           NET-HTTP-Activation
Web-Lgcy-Mgmt-Console                 RPC-Over-HTTP-Proxy -Restart
```

If the server will be hosting the Hub Transport, Client Access, Mailbox, and Unified Messaging Server roles, then the following commands should be used:

```
Add-WindowsFeature NET-Framework      WAS-Process-Model
RSAT-ADDS                             RSAT-Web-Server
Web-Server                            Web-ISAPI-Ext
Web-Basic-Auth                        Web-Digest-Auth
Web-Windows-Auth                      Web-Dyn-Compression
Web-Metabase                          NET-HTTP-Activation
Web-Net-Ext                           RPC-Over-HTTP-Proxy
Web-Lgcy-Mgmt-Console                 Desktop-Experience -Restart
```

If your server will be configured with the Hub Transport and Client Access Server roles only, then this is the command that you would use:

```
Add-WindowsFeature NET-Framework      WAS-Process-Model
RSAT-ADDS                             RSAT-Web-Server
Web-Server                            Web-ISAPI-Ext
Web-Basic-Auth                        Web-Digest-Auth
Web-Windows-Auth                      Web-Dyn-Compression
Web-Metabase                          NET-HTTP-Activation
Web-Net-Ext                           RPC-Over-HTTP-Proxy -Restart
Web-Lgcy-Mgmt-Console
```

If the server will only host the Hub Transport and Mailbox Server roles, then you would use the following command:

```
Add-WindowsFeature NET-Framework      Web-Metabase
RSAT-ADDS                             Web-Net-Ext
Web-Server                            Web-Lgcy-Mgmt-Console
Web-Basic-Auth                        WAS-Process-Model
Web-Windows-Auth                      RSAT-Web-Server -Restart
```

Servers hosting the Client Access and Mailbox Server roles require the following commands:

```
Add-WindowsFeature NET-Framework      WAS-Process-Model
RSAT-ADDS                             RSAT-Web-Server
Web-Server                            Web-ISAPI-Ext
Web-Basic-Auth                        Web-Digest-Auth
Web-Windows-Auth                      Web-Dyn-Compression
Web-Metabase                          NET-HTTP-Activation
Web-Net-Ext                           RPC-Over-HTTP-Proxy -Restart
Web-Lgcy-Mgmt-Console
```

You would run the following commands to prepare dedicated Client Access Servers:

```
Add-WindowsFeature NET-Framework      WAS-Process-Model
RSAT-ADDS                             RSAT-Web-Server
Web-Server                            Web-ISAPI-Ext
Web-Basic-Auth                        Web-Digest-Auth
Web-Windows-Auth                      Web-Dyn-Compression
Web-Metabase                          NET-HTTP-Activation
Web-Net-Ext                           RPC-Over-HTTP-Proxy -Restart
Web-Lgcy-Mgmt-Console
```

The following command is used for preparing a dedicated Hub Transport Server or a dedicated Mailbox Server:

```
Add-WindowsFeature NET-Framework      Web-Metabase
RSAT-ADDS                             Web-Net-Ext
Web-Server                            Web-Lgcy-Mgmt-Console
Web-Basic-Auth                        WAS-Process-Model
Web-Windows-Auth                      RSAT-Web-Server -Restart
```

Dedicated Unified Messaging Servers can be prepared by using this command:

```
Add-WindowsFeature NET-Framework      Web-Net-Ext
RSAT-ADDS                             Web-Lgcy-Mgmt-Console
Web-Server                            WAS-Process-Model
Web-Basic-Auth                        RSAT-Web-Server
Web-Windows-Auth                      Desktop-Experience -Restart
Web-Metabase
```

Edge Transport Servers can be prepared by using the following command:

```
Add-WindowsFeature NET-Framework      ADLDS -Restart
RSAT-ADDS
```

Several of the commands listed above prepare the server for the deployment of a Client Access Server. Any time that you are deploying the Client Access Server role, you must enter the following command after running the initial command set and rebooting the server:

```
Set-Service NetTcpPortSharing -StartupType Automatic
```

Active Directory

Prior to installing Exchange Server 2010, the server on which Exchange is to be installed must be configured to be a domain member. The exception to this requirement is the Edge Transport Server role, which cannot be installed on a domain member.

If any domain controllers are running Windows 2000 Server or Windows 2000 Advanced Server, those domain controllers must be either upgraded to Windows Server 2003 or removed from the domain.

Prior to installing Exchange Server 2010, the Active Directory's forest Functional Level must be set to Windows Server 2003 or higher.

The Schema Master within the forest in which Exchange Server 2010 is being deployed must be running Windows Server 2003 SP1 or higher.

Every domain that will contain an Exchange 2010 server must have at least one domain controller running one of the following operating systems. In each case, the domain controller can run either the 32-bit or the 64-bit version of the operating system. The acceptable operating systems include:

- Windows Server 2003 Standard Edition with SP1 or higher
- Windows Server 2003 Enterprise Edition with SP1 or higher
- Windows Server 2008 Standard Edition
- Windows Server 2008 Enterprise Edition
- Windows Server 2008 R2 Standard Edition
- Windows Server 2008 R2 Enterprise Edition

The Active Directory must be prepared prior to installing Exchange 2010. If you have Schema Admin, Enterprise Admin, and Domain Admin permissions, then the Exchange Server 2010 Setup Wizard can prepare the Active Directory for you. However, you should expect to see some test questions asking you about the process for manually preparing the Active Directory.

If you will be using a computer that's running Windows Server 2008 when you prepare the Active Directory, then you must install the Active Directory management tools prior to modifying the Active Directory. You can install these tools by executing the following command:

```
ServerManagerCmd –I RSAT-ADDS
```

Prepare the Legacy Exchange Server Permissions

If you have an existing Exchange Server organization in place, and that organization contains servers that are running Exchange Server 2003, then the first step in preparing the Active Directory is to prepare the legacy Exchange permissions. Technically, you can skip this step and the legacy Exchange permissions will be prepared automatically as you prepare the Active Directory, but Microsoft expects you to know how to manually prepare the legacy Exchange Permissions.

To prepare the legacy Exchange Permissions for every domain in the entire forest which contains an Exchange Server, you must be a member of the Domain Admins group for each domain that you are preparing. The command for preparing legacy Exchange Server permissions can be executed on any computer that is running Windows Server 2008 or Windows Server 2008 R2, so long as it resides in the same domain and same site as the Schema Master.

The command used for preparing legacy Exchange Server Permissions is:

```
Setup /PrepareLegacyExchangePermissions
```

As a shortcut, you can use this command instead:

```
Setup /PL
```

If you want to prepare the legacy Exchange Permissions for a specific domain, then you would append the fully qualified domain name of the domain that you want to prepare. For example, if you had a domain named Contoso.com, then you would use this command to prepare it:

```
Setup /PrepareLegacyExchangePermissions:Contoso.com
```

Alternatively, you could use this command:

```
Setup /PL:Contoso.com
```

Preparing the Active Directory Schema

The second step in preparing the Active Directory is to prepare the schema. Once again, you can skip this step and Setup will perform it automatically. However, Microsoft expects you to know how to manually extend the Active Directory Schema.

In order to update the schema, you must be a member of the Schema Admins and the Enterprise Admins groups. Additionally, the command for extending the schema must be executed on a server that is running a 64-bit Windows operating system, and this computer must reside in the same domain and Active Directory site as the schema master.

To extend the Active Directory Schema, use the following command:

```
Setup /PrepareSchema
```

As an alternative, you can use this command:

```
Setup /PS
```

After you have extended the Active Directory Schema, you must wait for the changes to be replicated to your other domain controllers before moving forward with the rest of the Active Directory preparation process.

Completing the Active Directory Preparation

There is one last step in preparing the Active Directory. The actual command that you will use when you perform this step will vary depending on whether or not there is already an Exchange Server organization in place.

If your network does not already contain an Exchange organization, then you will have to come up with an organization name and include it in the command. The organization name must be 64 characters or less in length, and cannot be changed once it is set. If you have an existing Exchange Server organization in place, then your existing organization name will be used. The organization name can contain spaces, but you must enclose the name in quotation marks if you choose to use spaces.

To complete the preparation of the Active Directory, you must be a member of the Enterprise Admins group. The command must be executed on a server residing in the same Active Directory site and domain as the Schema Master. The computer on which you execute the command must also be able to contact every domain in the entire forest over port 389.

To complete the Active Directory preparation, you must use this command:

```
Setup /PrepareAD
```

As an alternative, you can use this command instead:

```
Setup /p
```

If you have neglected to prepare the legacy Exchange Server permissions or extend the Active Directory schema, then those tasks will be completed automatically when you run Setup /PrepareAD (assuming that you have the required permissions).

Performing a Typical Exchange Server Deployment from the GUI

Microsoft defines a typical Exchange Server 2010 deployment as a server that contains the Hub Transport, Client Access, and Mailbox Server roles, and the Exchange Management Tools. You can perform a typical Exchange Server deployment from the GUI by performing these steps:

1. Insert the Exchange Server installation DVD and run Setup.
2. When Setup displays the Welcome screen, click Next.
3. When Setup displays the Error Reporting page, choose to either enable or disable error reporting and click Next.
4. When you arrive at the Installation Type page, select the Typical Exchange Server Installation option. It is worth noting that selecting this option prevents you from adding any additional server roles, or removing any server roles until after the installation completes. Click Next to continue.
5. If this is the first Exchange Server that is being deployed in a new Exchange Server organization, and if you neglected to provide an organization name when you prepared the Active Directory, then Setup will prompt you to enter the organization name that you want to use.
6. If this is the first Exchange Server in a brand new organization, then you will see a screen asking you if you have client computers that are running Outlook 2003 or earlier versions of Outlook. If such clients exist on your network, then Setup will create a public folder database as part of the mailbox server deployment. If no Outlook 2003 clients exist, then a public folder database is not required.
7. The following screen will ask you if the Client Access Server role will be Internet facing. If you will be receiving Internet mail, then you should select the check box indicating that the Client Access Server will be Internet facing. Additionally, you must also specify your external domain name before you click Next.
8. You should now be taken to the Customer Experience Improvement page. Make your selection regarding customer experience improvement and click Next.
9. Setup should now perform a readiness check to ensure that all of the necessary prerequisite components are in place. You must correct any errors before continuing. You should also review any warnings that might be displayed.
10. Click the Install button to begin installing Exchange.
11. When the installation process completes, click Finish.

Installing Exchange in Unattended Mode

Just as you can install Exchange using the Setup wizard, you can also install Exchange from the command line using Unattended Mode. When using Unattended Mode, you must meet the same prerequisites as if you were going to be installing Exchange from the GUI.

Setting up Exchange in Unattended Mode involves using the Setup.com command in conjunction with various command line switches. The full command syntax is:
Setup.com [/mode:<setup mode>] [/role:<server roles to install>] [/OrganizationName:<name for the new Exchange organization>] [/TargetDir:<target directory>] [/SourceDir:<source directory>][/UpdatesDir:<directory from which to install updates>] [/DomainController:<FQDN of domain controller>] [/AnswerFile:<filename>] [/DoNotStartTransport] [/EnableLegacyOutlook] [/LegacyRoutingServer] [/EnableErrorReporting] [/NoSelfSignedCertificates] [/AdamLdapPort:<port>] [/AdamSslPort:<port>] [/LanguagePack:<language pack bundle>] [/AddUmLanguagePack:<UM language pack name>] [/RemoveUmLanguagePack:<UM language pack name>] [/NewProvisionedServer:<server>] [/RemoveProvisionedServer:<server>]

The first switch used in the syntax above is the Mode switch, which can be used to put the server into installation mode. The options that you can use include Install, Uninstall, and Recover Server. Install is the default action, and is used if you do not specify the Mode switch. Uninstall is used to remove Exchange Server roles. The RecoverServer switch is used in disaster recovery situations, and I will talk about it later in this manual.

The second switch is the Roles switch. You can use the Roles switch to specify the roles that you want to install or uninstall. For example, if you wanted to install Exchange as a Client Access Server, you could do so by using the following command:

```
Setup /Mode:Install /Roles:ClientAccess
```

Here is a list of the roles that you can install:

- HubTransport, or HT, or H
- ClientAccess, or CA, or C
- Mailbox, or MB, or M
- UnifiedMessaging, or UM, or U
- EdgeTransport, or ET, or E
- ManagementTools, or MT, or T
  (The Management Tools are installed by default with any other server role)

The next switch that you can use is /TargetDir. This switch is followed by the path to which you want to install Exchange. This is an optional parameter. If you do not specify a target folder, then Exchange will be deployed to the \Program Files\Microsoft\Exchange Server folder. Safeguards prevent the installation of Exchange to the root directory or to removable media.

The /SourceDir switch is an optional switch that allows you to specify the location of the Exchange Server installation files. If you do not specify a source directory, then Setup uses the current directory.

The /UpdatesDir switch allows you to specify the location of any updates that you want to include in your Exchange installation.

The /DomainController switch allows you to require Setup to use a specific domain controller when installing Exchange. This switch is not normally required. If you do use it, you must append the fully qualified domain name of a domain controller that is in the same Active Directory site as the computer on which you are running Setup.

The /AnswerFile parameter allows you to provide the location for an answer file that is to be used with Setup. Creating an answer file is an easy way to install multiple Exchange servers in a consistent manner.

By default, the transport service starts when Setup completes, but you can use the /DoNotStartTransport switch to prevent it from starting.

You should specify the /EnableLegacyOutlook switch if you are deploying a mailbox server and you have clients that are running Outlook 2003 or earlier. This switch will cause a public folder database to be created on the mailbox server.

You can enable error reporting with the /EnableErrorReporting switch.

You can prevent Exchange from creating a self-signed certificate by specifying the /NoSelfSignedCertificate switch. This will cause all communications to be unencrypted.
If you are deploying an Edge Transport Server, you should use the /AdamLdapPort switch to specify the LDAP port that will be used to access the server's directory partition. Microsoft recommends setting the port number to 50389 (/AdamLdapPort:50389).

Another switch that should be used when deploying an Edge Transport Server is the /AdamSslPort. The ADAM SSL port number should be set to 50636.

If you need to deploy a language pack for Exchange, you can provide the path to the language pack by using the /LanguagePack switch.

Unified Messaging servers require a different type of language pack, which should be specified through the use of the /AddUmLanguagePack switch. Likewise, Unified Messaging language packs can be removed by using the /RemoveUmLanguagePack switch.

The /NewProvisionedServer switch can be used to create an Active Directory object for a new Exchange Server without actually deploying Exchange. This allows one administrator to create a new server object in the Active Directory, and a different administrator to install Exchange. Likewise, provisioned servers can be removed from the Active Directory by using the /RemoveProvisionedServer switch.

Legacy Exchange Server Coexistence

Microsoft supports Exchange 2010 coexistence with Exchange Server 2007 and Exchange Server 2003. If an organization has any Exchange 2000 servers, those servers must be decommissioned or upgraded to Exchange 2003 prior to deploying Exchange 2010.

If you are going to deploy Exchange 2010 into an existing Exchange organization, that organization must be operating in native mode. Switching to Native Mode prevents any Exchange 2000 servers from operating within the organization.

You can switch Exchange 2003 to Native Mode by completing these steps:

1. Open the Exchange System Manager.
2. Right click on the Organization container and choose the Properties command from the shortcut menu.
3. When the resulting properties sheet opens, go to the General tab and then click the Change Mode button.
4. When asked if you want to change modes, click Yes.

Upgrading from Exchange 2003

You cannot perform an in place upgrade from Exchange 2003 to Exchange 2010. Instead, you will have to bring one or more Exchange 2010 servers into your existing Exchange Server organization and then migrate the data from Exchange Server 2003 to Exchange 2010. The migration is optional however, because long term coexistence between Exchange 2003 and Exchange 2010 is fully supported.

The first step in the upgrade process is to create a set of legacy hostnames that can be associated with your Exchange 2003 infrastructure. However, you can skip if you plan to migrate all of the mailboxes right away. Legacy hostnames are only required if you won't be able to move all of the mailboxes at once.

Legacy hostnames follow a specific format. The word legacy is attached to the existing hostname. For example, if your existing hostname is contoso.com, then the legacy hostname would be legacy.contoso.com.

Once you have created the DNS records for the new hostname, you will have to acquire a new SSL certificate that will support the legacy hostname. Microsoft recommends using an X.509 certificate with Subject Alternate Name (SAN) support, but the use of wildcard certificates is also supported.

When you are ready to begin deploying Exchange 2010, the first server role that you must deploy is the Client Access Server role. As you deploy the CAS role, you will be prompted to provide an external Client Access Domain. The domain name that you provide should be the domain name that was previously in use by your Exchange 2003 organization (such as Contoso.com, or mail.contoso.com).

If you plan to use Outlook Anywhere, then you can configure it at this point. The EMS command used for doing so is:

> Enable-OutlookAnywhere –Server:server_name –ExternalHostName:external_client_access_domain_name –SSLOffloading $false

If you deploy the CAS server by using the GUI, then you are prompted to provide the external Client Access Domain name. If you deploy the server from the command prompt, then the external Client Access Domain can be specified by using the /ExternalCASServerDomain switch. If you neglect to use this switch, Setup will complete, but Setup will not correctly configure the various virtual directories that are used by the CAS server. In this case, you will have to manually configure these directories later on by using the following commands:

Virtual Directory	Command
Offline Address Book	Set-OABVirtualDirectory <server_name>\OAB* -External URL https://contoso.com/OAB
Web Services	Set-WebServiceVirtualDirectory <server_name>\EWS* -ExternalURL https://contoso.com/exchange.asmx
ActiveSync	Set-ActiveSyncVirtualDirectory –Identity <servername>\Microsoft-Server-ActiveSync –ExternalURL https://contoso.com
Outlook Web App	Set-OWAVirtualDirectory <server_name>\OWA* -ExternalURL https://contoso.com/OWA
Exchange Control Panel	Set-ECPVirtualDirectory <server_Name>\ECP* -ExternalURL https://contoso.com/ECP

Although the legacy hostnames should have already been created, you must make the CAS server aware of the legacy hostnames in order for Exchange 2003 and Exchange 2010 to coexist. The command for doing so is:

```
Set-OWAVirtualDirectory <server_name>\OWA* -Exchange2003URL https://
legacy.contoso.com/OWA
```

You must move the OAB generation server to the Exchange 2010 Server. To perform the move and designate the Exchange 2010 server as the Web distribution point for the OAB, enter the following commands into the Exchange Management Shell:

```
Move-OfflineAddressBook "Default Offline Address Book" -Server <server_name>
$OABVDir=Get-OABVirtualDirectory -Server <server_name>
$OAB=Get-OfflineAddressBook "Default Offline Address List"
$OAB.VirtualDirectories and $OABVDir.DistinguishedName = Set-
OfflineAddressBook "Default Offline Address List" -VirtualDirectories
$OAB.VirtualDirectories
```

You must enable Integrated Authentication on the Microsoft-Server-ActiveSync virtual directory located on the backend Exchange 2003 server. Prior to doing so, you will have to install the patch that is associated with Microsoft Knowledgebase article 937031. You should use the Exchange System Manager to make this change. If you attempt to modify the authentication level through the IIS Manager, then Exchange will undo your changes.

The last step in the CAS server deployment process is to redirect the external MX record to point to the Exchange 2010 CAS server.

Upgrading Transport Servers

If you have Exchange 2007 running in your organization, then you must upgrade your CAS server first (using the same method that is used for Exchange 2003), and then upgrade your Hub Transport Servers and Edge Transport Servers.

The first step in preparing to upgrade the transport is to ensure that all existing hub and edge transport servers are running Exchange Server 2007 SP2.

If you have Exchange 2007 hub transport servers in multiple sites, then you must upgrade the Internet facing site first.

After you have deployed your first Exchange 2010 hub transport server, it will take over the communications with your Edge transport server. However, you will have to re-subscribe your edge transport server. If you plan to deploy multiple Exchange 2010 hub transport servers, then it is better to wait until you have deployed all of your new Exchange 2010 hub transport servers before subscribing the edge server.

Once the Exchange 2010 hub transport servers are in place, you should deploy your Exchange 2010 Edge Transport Server and subscribe it.

When the Exchange 2010 Edge Transport Server is up and running, you can remove the Exchange 2007 edge subscription, and decommission your Exchange 2007 Edge Transport Server.

Any Active Directory site containing both Exchange 2007 and Exchange 2010 servers must have both an Exchange 2007 and an Exchange 2010 hub transport server. This requirement is due to the fact that the Exchange 2007 and Exchange 2010 transport pipelines are not compatible with each other. Exchange gets around this compatibility problem by using versioned routing. Versioned routing works by checking the Exchange Server version that is running on a recipient's mailbox server, and then routing messages to that server through the appropriate hub transport server.

Exchange 2003 Mailbox Servers

If you plan to have both Exchange 2003 and Exchange 2010 mailbox servers coexisting (even for a short period of time) then you will need to suppress link state updates. Otherwise, routing loops can occur.

Suppressing link state updates involves editing the Exchange 2003 server's registry. If you make a mistake while editing the registry, you can destroy Windows and/or Exchange. You should therefore make a full system backup prior to performing any registry edit.

You can suppress link state updates by going to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\RESvc\Parameters and creating a new DWORD value named SuppressStateChanges. You should set the value's data field to 1.

Moving Mailboxes

You must use Exchange 2010's management tools when you move mailboxes from a legacy Exchange Server to an Exchange 2010 server. This means that you can't use the Exchange System Manager or Active Directory Users and Computers to move messages from an Exchange 2003 mailbox server to an Exchange 2010 mailbox server. Likewise, you can't use the Exchange 2007 Move-Mailbox cmdlet to move mailboxes to an Exchange 2010 server.

Exchange 2010 uses a new mechanism for moving mailboxes called Move Requests. Move Requests are handled by the Mailbox Replication Service. Because of this, the Exchange Management Shell and the Exchange Management Console are not directly involved in the mailbox move. It is possible to issue a move request and then close the management tool. In Exchange 2007, the management tool had to remain open until the move completed.

The primary advantage to using Move Requests is that the user's mailboxes remain online during the move. Online moves are only supported when moving mailboxes between Exchange 2007 SP2 and Exchange 2010 or between two Exchange 2010 servers.

Another advantage to Move Requests is that the Content Indexing feature begins scanning the mailbox contents as soon as the move begins. That way, mailbox items are searchable as soon as the move completes.

When you move a mailbox, the mailbox's move history is retained as a mailbox attribute. Likewise, recoverable items are moved with the mailbox.

In Exchange 2007, you could move mailboxes using the Move-Mailbox command. In Exchange Server 2010, you use the New-MoveRequest command instead.

Remote Mailbox Moves

In Exchange 2010, you can move mailboxes across Active Directory forest boundaries. These moves are known as Remote Mailbox Moves.

Remote Mailbox Moves are supported regardless of whether or not the remote forest contains an Exchange 2010 Client Access Server. However, in the absence of an Exchange 2010 Client Access Server in the remote forest, you will not be able to use the Exchange Management Console to perform the remote mailbox move. You will only be able to complete the move by using the Exchange Management Shell.

Prior to performing a cross forest mailbox move, you must create a mail enabled user in the target forest.

Microsoft recommends using Identity Lifecycle Management (ILM) for cross forest Global Address Synchronization. Feature Pack 1 for ILM 2007 contains sample code that you can adapt to get ILM to synchronize the source and target mailboxes.

ILM 2007 is not required for cross forest mailbox moves. If an organization does not have ILM in place, then they can use the UpdateRecipient cmdlet to generate the LegacyExchangeDN for the target mailbox. You can automate this process by using the Prepare-MoveRequest.ps1 PowerShell script.

You can use move requests to push a mailbox from an Exchange 2010 source forest to a remote Exchange 2007 or 2003 forest. The command for doing so looks like this:

```
New-MoveRequest -Identity 'JohnDoe@contoso.com -RemoteLegacy
-RemoteTargetDatabase DB01 -RemoteGlobalCatalog 'GC.contoso.com'
-RemoteCredential $Cred -TargetDeliveryDomain 'northwindtraders.com'
```

Typically, a remote move request will be issued from the target forest. This is what the move request might look like in such a situation:

```
New-MoveRequest -Identity 'JohnDoe@contoso.com' -TargetDatabase DB1
-RemoteHostName 'CAS.contoso.com' -RemoteCredential (Get-Credential
domain\Administrator) -TargetDeliveryDomain 'northwindtraders.com'
```

If the source forest contains legacy Exchange 2003 or Exchange 2007 mailboxes, then you must provide the mailbox identity, the RemoteLegacy switch, the FQDN of a remote global catalog server, the external e-mail address that will be created, and the name of the target database. An example of the command used to perform this type of move request is:

```
New-MoveRequest -Identity 'JohnDoe@contoso.com -RemoteLegacy
-TargetDatabase DB01  -RemoteGlobalCatalog 'GC.contoso.com'
-RemoteCredential $Cred -TargetDeliveryDomain 'northwindtraders.com'
```

Domain 2: Configure Exchange Recipients and Public Folders

Recipients

Exchange Server 2010 supports the use of many different types of recipients. As such, it is important for you to be familiar with the recipient types that are available, as well as the purpose of each recipient type. The table below illustrates the recipient types that are supported.

Recipient Type	Description
User Mailbox	A user mailbox is just a basic mailbox that is hosted on an Exchange 2010 server.
Legacy Mailbox	A legacy mailbox is similar to a user mailbox, except that the mailbox is hosted on a server that is running Exchange Server 2003.
Equipment Mailbox	An equipment mailbox is a mailbox that is designed to represent a specific piece of portable equipment. Users can use the equipment mailbox's calendar to reserve the corresponding piece of equipment. For example, you might create an equipment mailbox that can be used to reserve an LCD projector or a digital whiteboard.
Room Mailbox	A room mailbox is similar to an equipment mailbox, except that it is designed to allow users to reserve a conference room.
Linked Mailbox	A linked mailbox is a mailbox that belongs to a user within an external forest.
Mail Contact	A mail contact is an Active Directory object that points to an external user with a foreign SMTP address.
Mail Forest Contact	A mail forest contact is a read only, Active Directory object that is typically created by Microsoft's Identity Integration Service. It is designed to represent the identity of a user from an external forest.
Microsoft Exchange Recipient	Microsoft Exchange Recipient is a special mailbox that Exchange Server 2010 creates automatically. It is used for internal mail delivery. For example, the Journaling feature makes use of the Microsoft Exchange Recipient mailbox to send messages to the Journal mailbox.
Shared Mailbox	A shared mailbox is a mailbox that is shared among multiple users.
Linked User	A linked user is a user whose account exists in one forest while their mailbox resides in another. This is not to be mistaken for a linked mailbox.
Archive Mailbox	In Exchange 2010, each mailbox can have an accompanying archive mailbox in which users can deposit messages that they wish to keep for a period of time that exceeds the retention settings that have been applied to the user's normal mailbox.
Mail Enabled Universal Security Group	A mail enabled universal security group is a special type of distribution group that can be used to establish permissions to access a resource, but that can also be used to send messages to group members.

Mail Enabled Universal Distribution Group	A mail enabled universal distribution group is similar to a mail enabled universal security group, except that it cannot be used to provide access to resources. It is only used for distributing messages to group members.
Mail Enabled Non-Universal Group	A mail enabled non universal group is a legacy distribution group. These groups cannot be created in Exchange Server 2010, and their existence is only supported if they previously existed on an Exchange 2003 server.
Mail Enabled Public Folder	A mail enabled public folder is a public folder that has an e-mail address attached to it. Users can post content to the folder by sending a message to the corresponding e-mail address.
Dynamic Distribution Group	A dynamic distribution group is a special type of distribution group whose membership is determined at the time that a message is sent to the group, based on various filtering criteria.

Creating New Mailboxes

Microsoft generally recommends that you create an Exchange Server mailbox and the account that the mailbox is linked to simultaneously, although it is certainly possible to create a mailbox for an existing user account.

You can create a new mailbox by using either the Exchange Management Console or the Exchange Management Shell. To create a mailbox using the Exchange Management Console, navigate through the console tree to Recipient Configuration | Mailbox. Next, click the New Mailbox link, located in the Actions pane. When you do, Exchange will launch the New Mailbox Wizard.

The wizard's initial screen asks you what type of mailbox you would like to create, as shown in **Figure 1**. You have the option of creating a User Mailbox, Room Mailbox, Equipment Mailbox, or a Linked Mailbox. Make your selection and click Next.

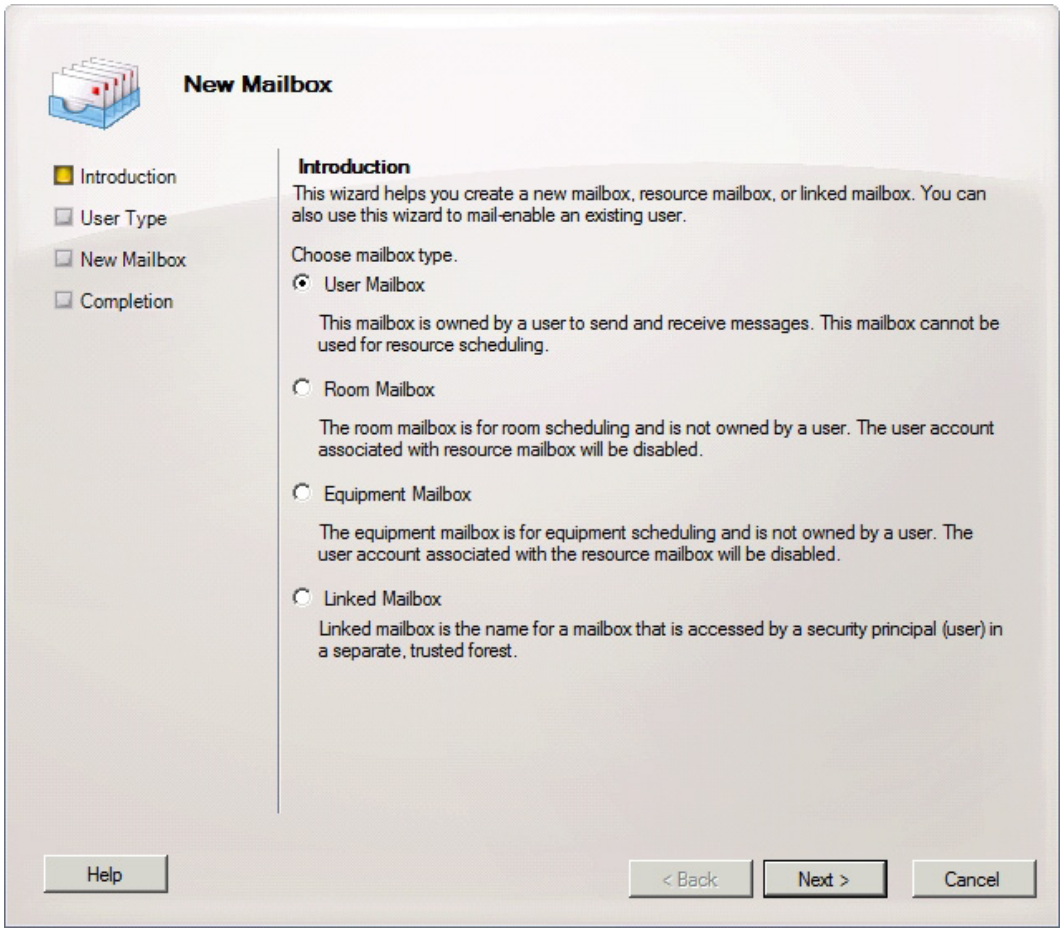

Figure 1: Creating a New Mailbox

The wizard's initial screen asks you what type of mailbox you would like to create.

At this point, you will see a screen asking you if you want to create a mailbox for a new user or for an existing user. If you choose to create a mailbox for a new user, then the wizard will lead you through a couple of additional screens that prompt you to provide details such as the name to be used for the new user account and the account's password. If you prefer to use an existing user account, then click the Add button, select the account that you want to use, and click Next.

The wizard's following screen prompts you for the user's alias. The alias is filled in automatically, and you won't typically have to make any changes to it.

This same screen also gives you the option of specifying the mailbox database in which you want to create the new mailbox. If you don't make a selection, then Exchange will choose a database automatically.

You also have the option of selecting a retention policy and an ActiveSync mailbox policy to be used with the mailbox. Again, these are optional and you do not have to specify policies if you do not want to.

After making your selections, click Next and you will see a summary page for the mailbox that you are about to create. If all seems to be correct, then click the New button to create the mailbox. When the process completes, click Finish.

Just as you can create a mailbox by using the Exchange Management Console, you can also create a mailbox from the Exchange Management Shell. The command used in doing so is Enable-Mailbox.

At a minimum, the Enable-Mailbox cmdlet requires you to provide the identity of the user account with which the mailbox will be associated, and the user's alias. Here is an example of such a command:

```
Enable-Mailbox –Identity 'Domain1.com/Users/User1' –Alias 'User1'
```

The command can be slightly more complex if you include details such as the name of the mailbox database, retention policy, and ActiveSync Mailbox Policy. Here is an example of such a command:

```
Enable-Mailbox –Identity 'Domain1.com/Users/User1' –Alias 'User1' –Database 'DB1'
–RetentionPolicy 'Default Archive Policy' –ActiveSyncMailboxPolicy 'Default'
```

In both of the previous examples, the Enable-Mailbox cmdlet was used to create the mailbox. It is important to note that the Enable-Mailbox cmdlet is only used if you are creating a mailbox for a user account that already exists. If you are creating a mailbox for a new user account, then you will use the:

```
New-Mailbox cmdlet instead.
```

As you would probably assume, the New-Mailbox cmdlet requires you to provide information related to the account that you are creating. Here is an example of what such a command might look like:

```
New-Mailbox –Name 'User1' –Alias 'User1' –UserPrincipalName 'User1@Domain1.
com' –SamAccountName 'User1' –FirstName 'User1' –Initials '' –LastName '' –
Password 'System.Security.SecureString' –ResetPasswordOnNextLogon $True
```

Room and Equipment Mailboxes

All of the examples in the previous section revolved around creating user mailboxes. However, you also need to know how to create resource mailboxes such as room mailboxes and equipment mailboxes.

A room or an equipment mailbox can be created through either the Exchange Management Console or the Exchange Management Shell. The process for creating an equipment mailbox is very similar to the process used in creating a user mailbox.

If you want to create a room or an equipment mailbox through the Exchange Management Console, then navigate through the console tree to Recipient Configuration | Mailbox. Next, click on the New Mailbox link, located in the Actions pane. This will cause Exchange to launch the now familiar New Mailbox Wizard.

As was the case before, the initial screen asks you to choose a mailbox type. This time, you should choose either the Room Mailbox or the Equipment Mailbox option. You will now be prompted as to whether you want to create a mailbox for a new user or an existing user. If you plan to use an existing user account, you must disable the account prior to creating the new mailbox.

If you decide to use an existing user account then click the Add button, select the account that you want to use, and click Next. Otherwise, click Next, and follow the prompts to create the new account. In either case, you should eventually end up at the Mailbox Settings screen.

As was the case before, this screen gives you the option of specifying an alias for the mailbox, as well as specifying a mailbox database, a retention policy, and an ActiveSync Mailbox Policy. Keep in mind that you won't typically use an ActiveSync mailbox policy with an equipment or a room mailbox.

Click Next and you will be taken to the summary screen for the new mailbox. Assuming that everything shown on this screen appears to be correct, then click New to create the mailbox. When the process completes, click Finish.

As was the case with a user mailbox, you can also use the Exchange Management Shell to create an equipment or a room mailbox. Just as before, you will use the New-Mailbox cmdlet if you are creating a new user account to go along with the mailbox. Otherwise, you will use the Enable-Mailbox cmdlet to create a mailbox for an existing user account.

The syntax for creating a room or an equipment mailbox is nearly identical to that of creating a user mailbox, but there is one important difference. To see that difference, check out the sample commands shown below:

```
Enable-Mailbox -Identity 'Domain1.com/Users/Projector' -Alias 'Projector' -Equipment
Enable-Mailbox -Identity 'Domain1.com/Users/Conference Room' -Alias 'Conference' -Room
```

As you can see, the first command used the –Equipment switch as a way of telling Exchange that an equipment mailbox is being created. Likewise, the second command used the –Room switch to tell Exchange that the new mailbox was to be a room mailbox.

Resource Properties

Sometimes allowing a user to simply reserve a piece of equipment or a conference room isn't enough. For example, imagine that a newly employed manager needs to book a conference room for a meeting with his staff. If the manager were to randomly select from the available conference rooms, he might accidentally choose a conference room that is located in a different building – maybe even in a different state. Likewise, if the employee is unfamiliar with the organization's facilities, he may book a conference room with a capacity of 500 people for a meeting of ten.

This is where resource properties come into play. Resource properties allow you to associate various attributes with room and equipment mailboxes. That way, you can give employees important information about a resource before they book it.

To provide resource information using the Exchange Management Console, navigate through the console tree to Recipient Configuration | Mailbox. Upon doing so, right click on the resource mailbox that you want to modify, and then choose the Properties command from the resulting shortcut menu. This will cause Exchange to display the properties sheet for the mailbox.

Go to the properties sheet's Resource General tab. In the case of a Room Mailbox, there will be a field in which you can enter the room's capacity. There is also a Resource Capacity field for equipment mailboxes, as shown in **Figure 2**.

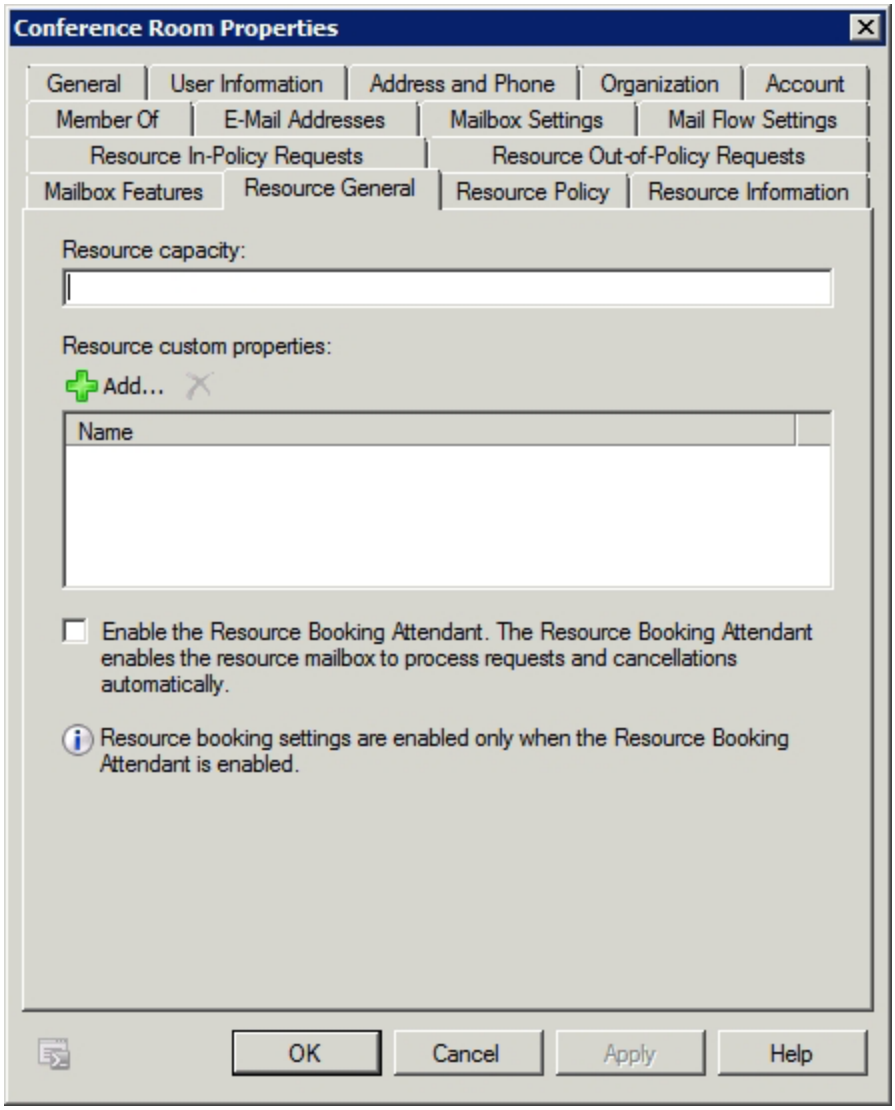

Figure 2: Specifying Resource Capacity

You can specify a room's capacity through the Resource General tab.

In either case, the tab also includes a section labeled Resource Custom Properties. You won't be able to define custom resource properties through the Exchange Management Console. Instead, you will have to use the Exchange Management Shell's Set-ResourceConfig cmdlet to extend the resource property schema.

Resource Scheduling

In an effort to reduce the overall administrative burden, Microsoft has designed Exchange Server 2010 so that when a user attempts to reserve a resource such as a conference room or a piece of equipment, the reservation can be automatically approved. Of course you probably don't want for just any user to be able to send meeting requests to a resource mailbox.

Exchange allows you to control which users can send meeting requests to resource mailboxes by using either the Exchange Management Console or the Exchange Management Shell. To use the Exchange Management Console, navigate through the console tree to Recipient Configuration | Mailbox. Next, right click on the resource mailbox for which you want to set the meeting request permissions, and then choose the Properties command from the resulting shortcut menu. Upon doing so, Exchange will display the mailbox's properties sheet.

At this point, go to the properties sheet's Resource In-Policy Requests tab, shown in **Figure 3**. This tab controls what happens to meeting requests that don't violate any resource scheduling options. For example, a meeting request might be considered to be in policy if it does not conflict with another meeting.

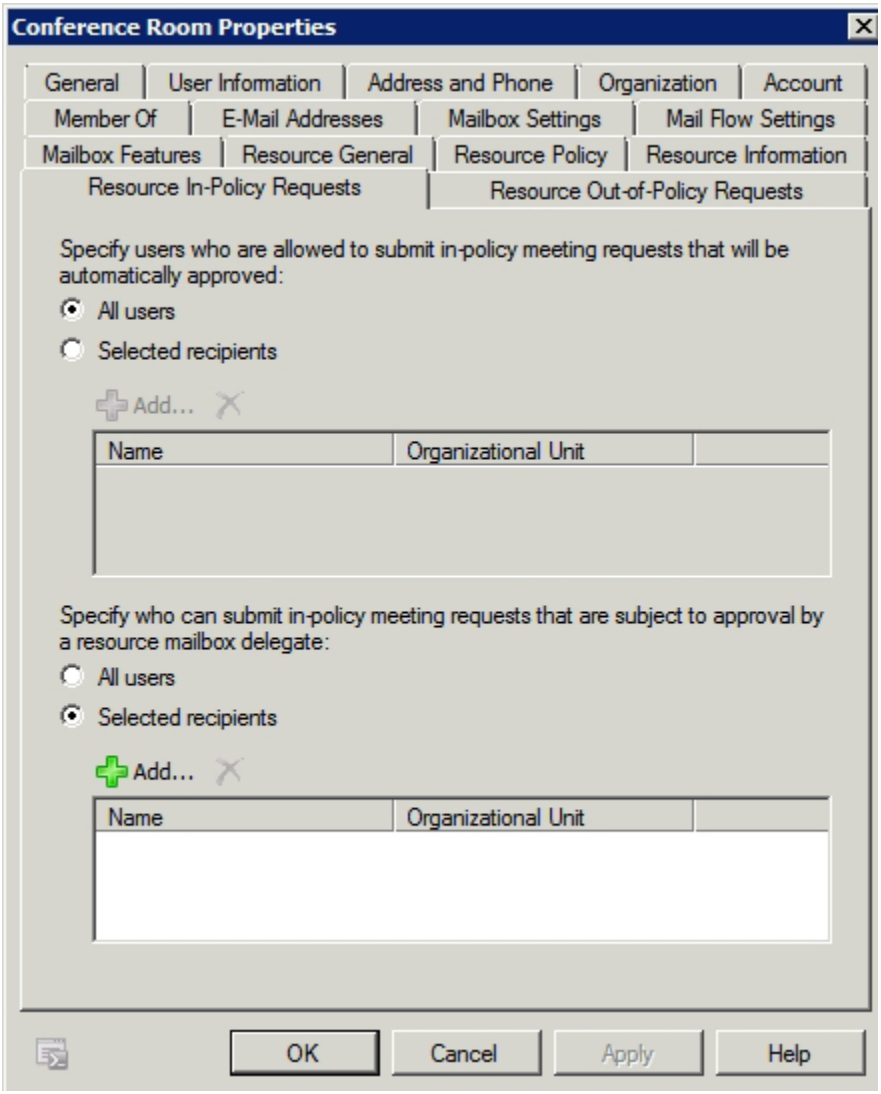

Figure 3: Configuring Resource In-Policy Requests

You can control automatic approvals on a per user basis.

Once you arrive on the Resource In-Policy Requests tab, you must choose either the All Users option or the Selected Recipients option. The All Users option allows everyone to send meeting requests to the mailbox, while the Selected Recipients option allows you to control who is allowed to send meeting requests.

When you are done, go to the properties sheet's Resource Out-of-Policy Requests tab. This tab controls who will be allowed to send out of policy requests to the resource mailbox. For example, suppose that a conference room has already been booked. A meeting request for that same time slot would be considered to be an out of policy request. If the user who is making the request has been specified on the Out-of-Policy Request tab, then the request will not be automatically denied. However, it will not be approved either. Instead, the request will have to be handled by a delegate for the resource mailbox.

As was the case with the In-Policy Requests tab, you have the option of granting Out-of-Policy requests to all users or to select users. It is generally advisable to grant Out-of-Policy Requests only to select users (perhaps the management team) to avoid potentially burdening the mailbox delegate with having to resolve numerous out of policy requests.

Just as you can use the Exchange Management Console to configure how Exchange will respond to meeting requests made of resource mailboxes, you can also use the Exchange Management Shell. You can configure the way in which resource mailboxes respond to meeting requests by using the Set-CalendarProcessing cmdlet.

There are several parameters that can be used with the Set-CalendarProcessing cmdlet. Some of these parameters include:

- **Identity** – This is where you specify the name of the resource mailbox for which you want to perform the action.
- **AutomateProcessing** – This parameter can be used to tell Exchange to automatically accept meeting requests that are sent to the resource mailbox.
- **AllRequestInPolicy** – This switch tells Exchange that all in policy requests should be made subject to approval.
- **AllRequestOutofPolicy** – This switch tells Exchange that out of policy requests should be subject to approval.
- **BookInPolicy** – The BookInPolicy switch indicates that in policy requests should be automatically approved.
- **RequestInPolicy** – This switch is similar to the AllRequestInPolicy switch, except that it is used to allow policy requests for specific users rather than for everyone.
- **RequestOutOfPolicy** – The RequestOutOfPolicy switch is similar to the AllRequestOutOfPolicyswitch, except that it allows you to specify specific users who should be allowed to make out of policy requests, rather than allowing those requests to be made by everyone.
- **ProcessExternalMeetingMessages** – The ProcessExternalMeetingMessages parameter controls whether or not meeting requests from users outside of the Exchange organization will be accepted.

Now that you have seen the various switches that can be used in conjunction with the Set-CalendarProcessing cmdlet, let's take a look at some examples of how this cmdlet is used. For the sake of demonstration, let's pretend that you need to configure the way that Exchange responds to meeting requests for a room mailbox named Conference. The commands used to configure automatic approvals for an equipment mailbox would work in exactly the same way.

With that in mind, imagine that you wanted to automatically approve all in policy meeting requests for the Conference mailbox. To do so, you would use the following command:

```
Set-CalendarProcessing –Identity "Conference" –AutomateProcessing
AutoAccept $true –AllBookInPolicy $true
```

Now, imagine that we still wanted to automatically approve in policy requests for the room, but that we only wanted to allow a user named JohnDoe to have their requests automatically approved. To do so, we could use the following command:

```
Set-CalendarProcessing –Identity "Conference" –AutomateProcessing
AutoAccept $True –BookInPolicy "JohnDoe@Domain1.com"
```

Notice how this command uses BookInPolicy rather than AllBookInPolicy. This is what prevents Exchange from automatically approving requests from everyone.

Similarly, we can configure Exchange to accept requests from everyone, but stipulate that all requests are subject to approval. To do so, we would use this command:

```
Set-CalendarProcessing –Identity "Conference" –AutomateProcessing
AutoAccept $True –AllRequestInPolicy $True
```

As mentioned earlier, you may have some users who need to be able to make out of policy requests. Out of policy requests will always be subject to approval by a delegate, and cannot be approved automatically. However, you can control who is allowed to submit out of policy requests by using a command similar to this one:

```
Set-CalendarProcessing –Identity "Conference" –AutomateProcessing
AutoAccept $True –RequestOutOfPolicy "JohnDoe@Domain1.com"
```

Finally, it is usually advisable to configure Exchange to reject meeting requests from the outside world. That way, you won't have to worry about a denial of service attack in which someone on the outside fills up your resource mailbox calendars with bogus appointments. You can reject external calendar requests for a resource mailbox by using a command similar to this one:

```
Set-CalendarProcessing –Identity "Conference" –ProcessExternalMeetingMessages $false
```

Notice that the command shown above is processed on a per mailbox basis. You can disable external meeting requests for a mailbox without doing so on an organization wide basis.

Setting Resource Mailbox Delegates

As you learned in the previous section, there are some situations in which resource mailboxes are unable to automatically approve meeting requests. As such, you will usually need to assign a delegate who can manually intervene in these types of situations.

You can use either the Exchange Management Console or the Exchange Management Shell to set a delegate on a resource mailbox. To use the Exchange Management Console, navigate through the console tree to Recipient Configuration | Mailbox Next, right click on the resource mailbox that you want to configure, and choose the Properties command from the resulting shortcut menu. When you do, Exchange will display the mailbox's properties sheet.

Now, go to the properties sheet's Resource Policy tab, and then locate the Specify Delegates of This Mailbox section. Click the Add button and then specify the delegate for the mailbox.

It is also worth noting that near the bottom of this tab is a check box that you can use to forward meeting requests to the mailbox's delegates. If you are going to have a mailbox delegate, then it is usually a good idea to select this check box. When you have finished configuring the mailbox delegates, click OK.

If you want to specify a mailbox delegate by using the Exchange Management Shell, then you can do so by using the Set-CalendarProcessing cmdlet just as you did in the previous section. The only difference is that you will use the –ResourceDelegates switch to specify the delegates for the mailbox. For example, suppose that you wanted to make a user named JohnDoe a delegate for the Conference mailbox. To do so, you would use the following command:

```
Set-Calendaring –Identity "Conference" –ResourceDelegates "JohnDoe"
```

Mailbox Quota Management

Exchange Server 2010 allows you to apply quotas to both mailbox and public folder databases. If you are applying a quota to all of the mailboxes or to all of the public folders in an entire database then you can use either the Exchange Management Console or the Exchange Management Shell. If you need to apply a quota to an individual mailbox or to an individual public folder, you will have to use the Exchange Management Shell to do so.

There are three different types of quotas that can be applied to mailboxes and public folders:

- **Issue Warning** – A quota that tells Exchange to send a warning message once a mailbox or a public folder reaches a certain size.
- **Prohibit Send** (known as Prohibit Post when applied to a public folder database) – A quota that prevents the mailbox or folder owner from sending messages or posting to the public folder until the mailbox or folder's size is reduced.
- **Prohibit Send and Receive** – A quota that prevents the mailbox owner from being able to send or receive messages until the mailbox or folder size has been reduced. There is no public folder equivalent to this type of quota.

When any of these quota limits are reached, a message is sent with high importance to the mailbox or folder owner (quota messages are not subject to quota limits). In the event that a mailbox is owned by a security group, then the quota message will be sent to every member of the group.

Applying Quotas to a Mailbox Database

You can apply quotas to a mailbox database by using either the Exchange Management Console or the Exchange Management Shell. To use the console, navigate through the console tree to Organization Configuration | Mailbox. Next, choose the database that you want to manage, and then click the Properties link. This will cause the console to display the properties sheet for the database. The quota related settings are located on the properties sheet's Limits tab. The quota settings include:

- Issue Warning at (**KB**)
- Prohibit Send at (**KB**)
- Prohibit Send and Receive at (**KB**)

You can use each of these settings to establish threshold values for the various quota limits.

Just as you can use the Exchange Management Console to create mailbox database quotas, so too can you use the Exchange Management Shell. Quotas are implemented through the use of the Set-MailboxDatabase cmdlet. This cmdlet requires you to supply the name of the database to which you are applying the quota, the quota type, and the quota size. The quota size must be expressed as an integer, and is entered in KB. For example, if you wanted to set a quota limit at 2 GB, then the value would be entered as 2097152. Here are a few examples of how you can set quota limits from the command line:

```
Set-MailboxDatabase "Mailbox Database Name" -IssueWarningQuota 2097152

Set-MailboxDatabase "Mailbox Database Name" -ProhibitSendQuota 2097152

Set-MailboxDatabase "Mailbox Database Name" -ProhibitSendReceiveQuota 2097152
```

In these examples, the quota sizes are all identical. In real life, however, the IssueWarningQuota would be the smallest, the ProhibitSendQuota would be a little bit higher, and the ProhibitSendReceiveQuota would be higher than that. If you need to remove a quota, you can replace the quota size with the word Unlimited.

Public Folder Database Quotas

Database level public folder quotas can be applied by using either the Exchange Management Console or the Exchange Management Shell. You can apply quotas to a public folder database by navigating through the Exchange Management Console to Organization | Mailbox, clicking on the Database Management tab, and then selecting the public folder database that you want to configure. Click the Properties link, and the console will display the public folder's properties sheet. The quota settings are located on the Limits tab.

The quota settings used by public folder databases are slightly different from those used in conjunction with mailbox databases. The quota settings include:

- Issue Warning at: (**KB**)
- Prohibit Post at (**KB**)
- Maximum Item Size

The process of applying a public folder database quota from the command line works nearly identically to the process of applying a quota to a mailbox. Instead of using the Set-MailboxDatabase cmdlet, however, you must use the Set-PublicFolderDatabase cmdlet. Below are some examples of how you can apply a quota to a public folder database:

```
Set-PublicFolderDatabase "Public Folder Database Name" -IssueWarningQuota 2097152
Set-PublicFolderDatabase "Public Folder Database Name" -ProhibitPostQuota 2097152
```

Role Based Access Control

In Exchange Server 2007, Microsoft provided limited options for assigning administrative control over the Exchange organization. In Exchange Server 2010, the Exchange 2010 permission model has been replaced by a new feature called Role Based Access Control.

There are two primary ways in which role based access control differs from the permissions model used in Exchange Server 2007. For starters, role based access control isn't just for administrators. Roles can be assigned to the end users as well. The other way in which role based access control differs from its predecessor is in the fact that the roles are designed to align with actual job responsibilities. This is intended to make it easy to know which role should be assigned to a particular administrator or end user.

Role Based Access Control provides three methods for assigning permissions: Management Role Groups, Management Role Assignment Policies, and Direct Role Assignments. Here is a quick summary of what these mechanisms do.

- **Management Role Groups** – Used for assigning management roles to administrators.
- **Management Role Assignment Policies** – Used to assign roles to end users. This allows for very granular permissions to be assigned to users or groups. For example, you might use a role assignment policy to allow users to modify their own contact information.
- **Direct Role Assignment** – Used to manually assign individual permissions. Microsoft discourages the use of direct role assignments, unless it is absolutely necessary. As such, direct role assignments are beyond the scope of this exam.

Management Role Groups

Management Role Groups are really nothing more than universal security groups. These groups are predefined, and designed to allow group members to perform various administrative tasks, without being assigned any excessive rights beyond those required to complete the task. The table below lists the Management Role Groups that are included with Exchange Server 2010, as well as the purpose of each group.

Management Role Group	Function
Organization Management	The Organization Management role group is the most powerful of all of the management role groups. An administrator who has been assigned this role group has full administrative control over the entire Exchange Server organization.
View Only Organization Management	The View Only Organization Management role is used primarily for training purposes. An administrator who is assigned the View Only Organization Management role can view any object in the entire Exchange Server organization, but does not have permission to perform any administrative actions. This allows new Exchange administrators to gain familiarity with the Exchange Management Console without the risk of the accidental administrative changes.
Recipient Management	When an administrator is assigned the Recipient Management role, they are given the ability to create or modify Exchange Server mailboxes (recipients).
Discovery Management	The Discovery Management role is a specialized role that is used only for organizations that are required to perform message retention. Administrators who have been assigned this role are allowed to perform e-discovery searches across multiple mailboxes.
UM Management	The UM Management role allows administrators to perform Unified Messaging related management tasks. For example, such an administrator is allowed to configure Unified Messaging dial plans.
Help Desk	The Help Desk role provides limited recipient management capabilities and is intended for support staff. Such a user can use Outlook Web App to modify things like a user's department or phone number.
Delegated Setup	An administrator who has been granted the Delegated Setup management role has the ability to deploy new Exchange Servers, so long as the Active Directory has been properly provisioned ahead of time.
Hygiene Management	The Hygiene Management role group provides administrators with the ability to manage Exchange Server's antivirus and anti-spam features. This role group can be useful even if an organization uses third party antivirus and anti-spam software because many of the third party products are designed with this role group in mind.
Server Management	An administrator who has been granted the Server Management role group is allowed to perform server level management of the Exchange organization. Essentially, such an administrator is allowed to perform configuration tasks, but is not allowed to manage recipients.
Public Folder Management	As the name implies, adding an administrator to the Public Folder Management role group provides them with the ability to manage public folder databases.
Records Management	The Records Management role group is usually reserved for those who are responsible for ensuring an organization's regulatory compliance. Administrators who are assigned this role are given the ability to configure message retention policies, create and manage message classifications, and to manage transport rules.

Public Folders

Although public folders were "deemphasized" in Exchange 2007, they continue to be fully supported in Exchange Server 2010. As such, you can expect to see some exam questions related to basic public folder functionality.

Creating a Public Folder Database

Because public folders are not created by default (unless you have Outlook 2003 clients), you will need to know how to create a public folder database. To do so, navigate through the console tree to Organization Configuration | Mailbox. Click on the New Public Folder Database link, located in the Action pane. When you do, the console will launch a wizard that walks you through the process of creating the new database.

The wizard's initial screen requires you to provide a name for the database, as well as specify which server the database will reside on. The following screen requires you to specify a database file path and a log folder path (although these fields are populated by default), as shown in **Figure 4**. There is also a check box that you can use to control whether or not the database should be mounted immediately after creation.

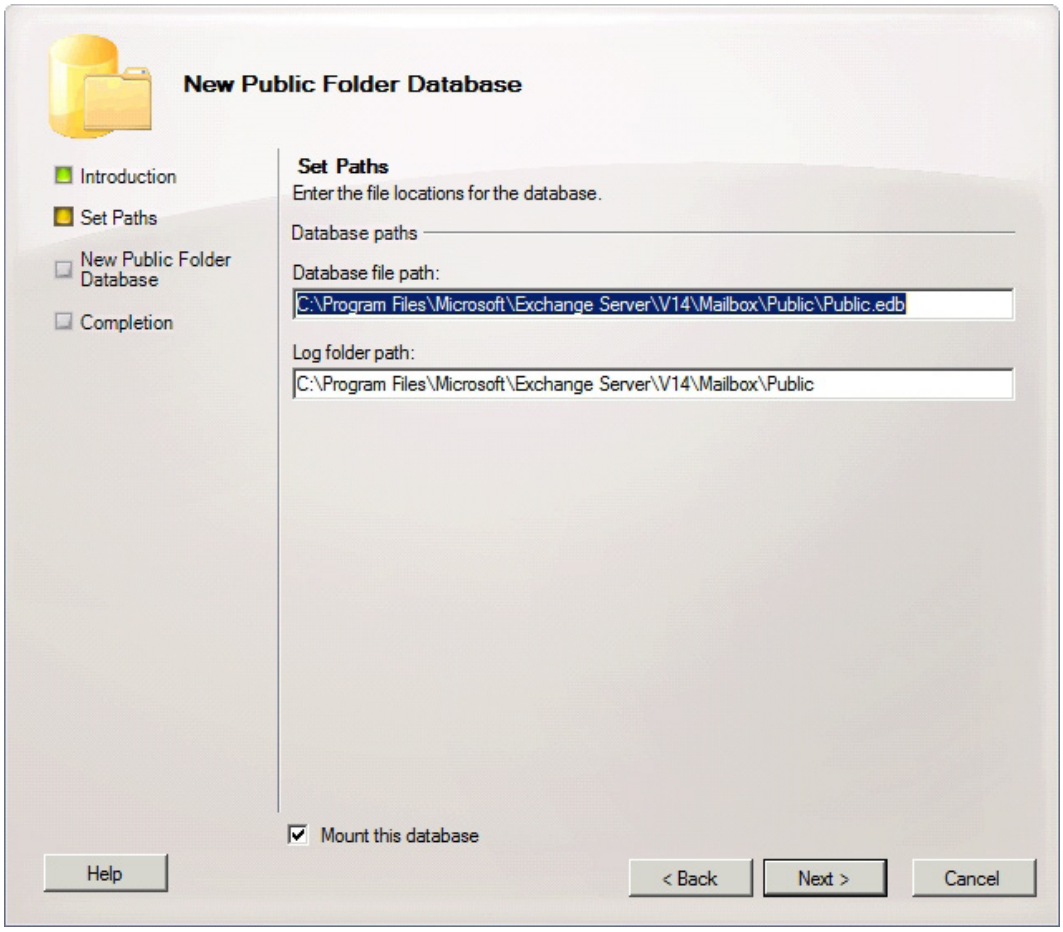

Figure 4: Creating a New Public Folder Database

The public folder paths are populated by default.

If you prefer, you can create public folder databases from the command line. To do so, you must use the New-PublicFolderDatabase command. At a minimum, you must supply a database name and the name of the server that will host the database. You can also provide the file and log paths if you like. Here is a sample of what the New-PublicFolderDatabase command looks like:

```
New-PublicFolderDatabase DatabaseName -Server SERVER1 -EdbFilePath E:\
Databases\Support\DatabaseName.edb -LogFilePath F:\Logs\Support
```

The command shown above creates a new database named DatabaseName on Server1, and also defines the name and location of the database file, as well as the location of the log files. You will notice that this command does not automatically mount the database. If you want to mount the database, you can do so by entering the Mount-Database command followed by the name of the database.

Creating a Public Folder

Public folders can be created through the Exchange Management Console and through the Exchange Management Shell. You can create a public folder using the console by clicking on the Toolbox container, and then choosing the Public Folder Management Console option. When the Public Folder Management Console opens, navigate to Default Public Folders. Next, select the top level folder beneath which you want to create the new folder, as shown in **Figure 5**. Finally, click the New Public Folder link, found in the Actions pane. When you do, the console will launch a wizard that you can use to create the new public folder.

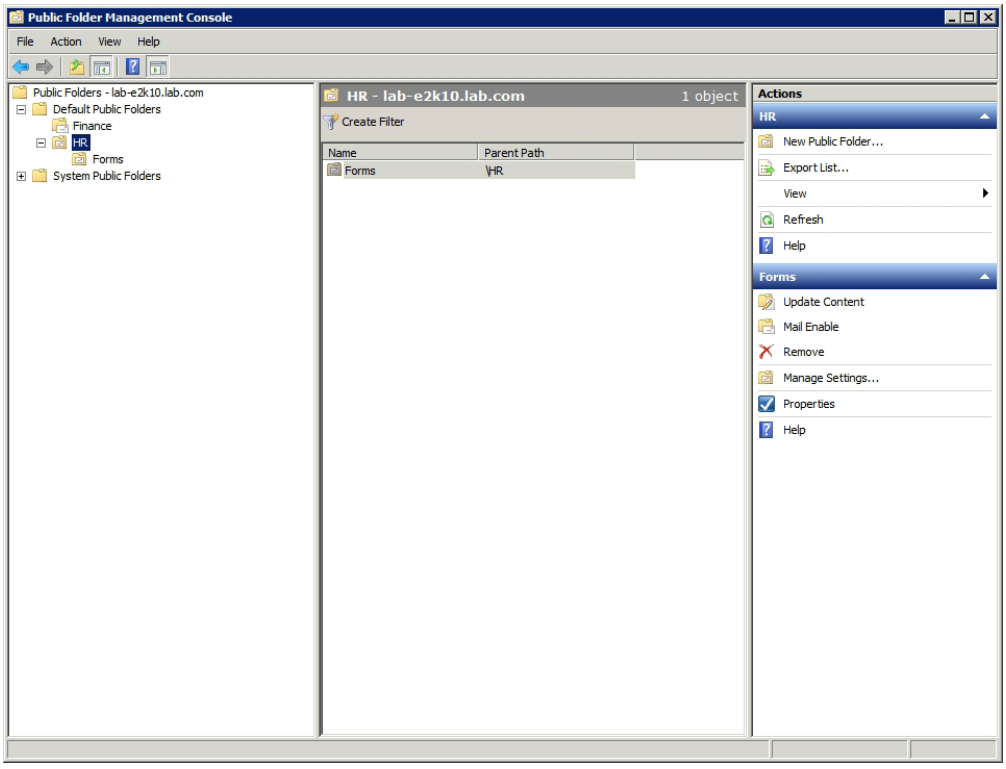

Figure 5: The Public Folder Management Console

Select the top level folder beneath which you want to create a new public folder.

The only information that you are required to provide is the name of the folder that you are creating. The wizard will display the public folder path, but this information is read only, and is designed solely as a means for verifying that you are creating the public folder in the correct location.

If you prefer to create a public folder from the Exchange Management Shell, you can do so by using the New-PublicFolder command. At a minimum, you only have to provide the name of the folder that you want to create. If you take this approach, then the folder will be created at the root level of the public folder tree on the closest public folder server.

If you require more control over the folder creation process, you can also specify the name of the server that will host the public folder, as well as the folder's path. Here is an example of the New-PublicFolder command:

```
New-PublicFolder –Name "My Public Folder" –Path \folders –Server "server1"
```

Public Folder Permissions

There are two main types of public folder permissions – client permissions and administrative permissions.

You can grant administrative permissions to a user by adding the user to a Public Folder Management role group Doing so is similar to assigning the Public Folder Administrator role in Exchange 2007 in that it grants the user all of the necessary permissions for managing the public folder. Using this method grants the user permission to manage the entire public folder tree. You can add a user account to the Public Folder Management role group by using this command:

```
Add-RoleGroupMember –Identity "Public Folder Management" –Member "John Doe"
```

If you need to assign more granular administrative control over the public folder structure, then you will need to use the PublicFolderAdministrativePermissions cmdlet to grant control to a specific public folder. Here is an example of how this cmdlet works:

```
Add-PublicFolderAdministrativePermission -Identity "\MyFolder" -User
"JohnDoe" -AccessRights AllExtendedRights -InheritanceType SelfAndChildren
```

Just as you need to know how to add administrative permissions to a public folder, you also need to know how to add client permissions. Microsoft generally recommends that you assign users predefined roles, but you can assign custom access rights, if necessary. To add an access right to a public folder, you will normally use the AddPublicFolderClientPermission cmdlet. Simply provide the cmdlet with the folder name, the user's name, and the rights that you want to assign. Here is an example of how to use this cmdlet:

```
Add-PublicFolderClientPermission –Identity "\folders\MyFolder" –
AccessRights PublishingEditor –User JohnDoe
```

The Add-PublicFolderClientPermission cmdlet works well if you need to grant a client permission to access a public folder. It is not the tool of choice if you need to grant permissions to a folder and all of the child folders beneath it. For that, you are better off using a PowerShell script that is included with Exchange. The name of the script is AddUsersToPFRecursive.ps1.

As is the case with the Add-PublicFolderClientPermission cmdlet, you must provide some additional parameters. Specifically, you must enter the name of the highest level public folder that you want to grant access to, the user's name, and the permission level. Here is an example of how to use the script:

```
AddUsersToPFRecursive.ps1 –TopPublicFolder "\MyFolder" –User "John Doe" –
Permission PublishingEditor
```

Mail Enabling Public Folders

A mail enabled public folder is a folder that has an e-mail address associated with it. When you mail enable a public folder, you allow users to post content to the folder by sending an e-mail message to the folder's e-mail address. You can mail enable a public folder by using either the Exchange Management Console or the Exchange Management Shell.

To use the console to mail enable a public folder, open the Exchange Management Console, go to the Toolbox, and click on the Public Folder Management Console. When the Public Folder Management Console opens, expand the Default Public Folder, and then select the folder that you want to mail enable. Once the folder is selected, click the Mail Enable link, located in the Actions pane. The folder's icon will change to reflect the fact that it has been mail enabled.

It is just as easy to mail enable a folder using the Exchange Management Shell. To do so, you must use the Enable-MailPublicFolder cmdlet. This cmdlet requires that you provide the folder's name, but you can also specify the name of the server that hosts the folder if necessary. Here is an example of the command that you would use:

```
Enable-MailPublicFolder –Identity "\MyFolder" –Server "Server1"
```

Public Folder Replication

Exchange Server 2010 allows you to create a replica of a public folder on an alternate mailbox server. The folders are kept synchronized according to a replication schedule. The replication schedule applies to the entire public folder database, but you can create a custom replication schedule that applies to a single public folder.

To create a public folder replica using the Exchange Management Console, open the console, go to the toolbox, and click on the Public Folder Management Console option. When the console opens, expand the Default Public Folders container, and then select the folder that you want to replicate. All of the child folders beneath the folder that you select will also be replicated.

With the folder selected, click on the Properties link found in the Actions pane to reveal the folder's properties sheet. Go to the properties sheet's Replication tab, and click the Add button. When prompted, choose the database that will house the replica folder and click OK. If you want to specify a custom replication schedule for the folder, then click the Customize button; otherwise click OK.

Public Folder Item Management

Just as you can apply quotas to a public folder, you can also apply retention limits that prevent folders from becoming congested with outdated items. You can set age limits at both the database level and at the individual folder level. By default, folders inherit the database level settings.

To apply age limits to a public folder database, open the Exchange Management Console and navigate to Organization Configuration | Mailbox. Next, go to the results pane, select the Database Management tab, and then select the public folder database that you want to configure. With the database selected, click on the Properties link found in the Actions pane. When the public folder database's properties sheet is displayed, go to the Limits tab.

When you specify the age limit for the items in the public folder, you must enter an integer corresponding to the number of days for which items should be retained. The valid range is between 0 and 24,855.

You can also use the Limits tab to set up deleted item retention for the database. The Keep Deleted Items For (Days) setting allows you to specify the number of days that deleted items should be retained within the public folder. The valid range is from 0 to 24,855 days.

Just below the Keep Deleted Items For (Days) setting is a check box labeled Don't Permanently Delete Items Until the Database Has Been Backed Up. As the name implies, this check box prevents the deleted items from being purged until a backup has been created.

Public Folder Management Scripts

Although you can fully manage public folders from the Exchange Management Shell, some management tasks can be rather complex. As such, Microsoft has included a number of PowerShell scripts with Exchange Server 2010 that are designed to make public folder management easier. These scripts fall into two basic categories – Administrative and User Management scripts. The table below contains a summary of the administrative scripts that are included with Exchange 2010.

Script Name	Function
AddReplicaToPFRecursive.ps1	This script can be used to create an additional replica of a public folder.
AggregatePFData.ps1	This script provides you with a summary of the statistics related to your public folders and the items within them.
RemoveReplicaFromPFRecursive.ps1	This script removes a replica of a public folder from a specified server.
MoveAllReplicas.ps1	You can use this script to remove all public folder replicas while creating duplicate replicas on another server.
ReplaceReplicaOnPFRecursive	This script is similar to the one listed above, except that it can be used to move a replica of a specific public folder (and its child folders) rather than moving the entire public folder hierarchy.

As mentioned a moment ago, Microsoft also provides several scripts for managing user access to public folders. The chart below provides a summary of these scripts and their purpose.

Script Name	Function
AddUsersToPFRecursive.ps1	This script adds a user and a specified set of permissions to a public folder and to all of the folders beneath it.
ReplaceUserWithUserOnPFRecursive.ps1	This script replaces one user with another on the access list for a public folder and for all of the folders beneath it. In doing so, the original user's permissions are applied to the new user.
ReplaceUserPermissionOnPFRecursive.ps1	This script allows you to replace a user's permissions with a different set of permissions. The change applies to the specified folder and to all child folders.
RemoveUserFromPFRecursive.ps1	This script revokes a user's permission to a public folder and to all of the folders beneath it.

Domain 3: Configuring Client Access

POP3 and IMAP4

Post Office Protocol 3 (POP3) is a legacy messaging protocol that is supported by Exchange 2010. The POP3 protocol is designed to download messages from a messaging server to a mail client such as Outlook. Once downloaded, the messages are removed from the server. As such, all messaging data resides solely on the client computer once it has been downloaded.

IMAP4 is a newer messaging protocol. Like POP3, it supports downloading messages to a client computer, but it can also be used for online messaging. Although IMAP4 is more advanced than POP3, it still lacks the ability to be used with Exchange Server's collaborative features such as calendars, contacts, etc.

POP3 and IMAP4 are both designed as mechanisms for transferring messages from a mail server to a mail client. They both lack the ability to send messages from the client to the server. Sending messages requires the use of the Simple Message Transport Protocol (SMTP).

Although Exchange 2010 fully supports POP3 and IMAP4, both protocols are disabled by default and are considered unnecessary in most Exchange Server environments. Microsoft provides POP3 and IMAP4 support as a way of offering universal compatibility with messaging clients other than Outlook. For example, if POP3 or IMAP4 are enabled, then users could connect to Exchange using a mail client such as Outlook Express or Entourage.

Even in cross platform environments, however, POP3 and IMAP4 are usually unnecessary because Exchange 2010 includes Outlook Web App – a Web based version of Outlook.

You can manage POP3 and IMAP 4 using either the Exchange Management Console or the Exchange Management Shell. To manage these protocols through the console, navigate through the console tree to Server Configuration | Client Access. Next, select the POP3 and IMAP 4 tab, located in the work pane. Finally, choose either the POP3 or IMAP4 option and then click the Properties link. This will open a properties sheet that you can use to manage POP3 and IMAP4 settings.

Managing POP3 and IMAP4 from the command line typically involves using the Set-PopSettings or the Set-ImapSettings commands. Likewise, you can also use the Get-PopSettings and Get-ImapSettings commands to retrieve information about the way that POP3 or IMAP4 are configured.

In addition to the server level configuration settings that are exposed through the Set-PopSettings and Set-ImapSettings commands, there are also commands available for managing POP3 and IMAP4 on an individual mailbox basis. Doing so involves the use of the Set-CASMailbox cmdlet.

The Set-CASMailbox cmdlet can be used for much more than just managing POP3 and IMAP4. For example, you can use the Set-CASMailbox command to enable Outlook Web App for a user. Even so, there are a very specific set of command parameters used for managing POP3 and IMAP4. These parameters include:

- **ImapEnabled** – This option controls whether or not IMAP4 is enabled for a particular mailbox.
- **ImapMessageRetrievalMimeFormat** – You can use this switch to control the format of messages retrieved from the user's mailbox.
- **ImapUseProtocolDefaults** – This option controls whether or not the default protocol settings that have been configured on the Client Access Server will be used for the IMAP4 protocol.
- **PopEnabled** – You can use this setting to enable or disable POP3 support.
- **PopMessageRetrievalMimeFormat** – This option specifies the format of messages that are retrieved from the mail server using the POP3 protocol.
- **PopUseProtocolDefaults** – This option controls whether or not the default protocol settings that have been configured on the Client Access Server will be used for the POP3 protocol.

An example of how to use the Set-CASMailbox cmdlet to manage POP3 or IMAP4 is:

```
Set-CASMailbox –Identity User1@Domain1.com –PopEnabled:$True
```

This command enables the POP3 protocol for User1@Domain1.com.

Protocol Logging

Occasionally, you may find yourself having to troubleshoot POP3 or IMAP4 problems. When these situations arise, the best way to begin the troubleshooting process is by enabling protocol logging.

Protocol logging for POP3 and IMAP4 is typically done from the Exchange Management Shell by using the Set-IMAPSettings or the Set-PopSettings commands. To enable protocol logging, use the following commands:

```
Set-IMAPSettings -protocolLogEnable –true
Set-POPSettings - protocolLogEnable –true
```

Protocol logging will not go into effect until you start and stop the underlying services. To restart the POP3 service, use the following commands:

```
Net Stop MSExchangePOP3
NetStart MSExchangePOP3
```

To restart the IMAP4 service, you can use these commands:

```
Net Stop MSExchangeIMAP4
Net Start MSExchangeIMAP4
```

As an alternative, you can enable, disable, or modify protocol logging by editing the Microsoft.Exchange. POP3.exe.config or the Microsoft.Exchange.Imap4.exe.config files. However, Microsoft typically recommends using the Set-POPSettings and the Set-IMAPSettings commands rather than directly modifying configuration files.

The protocol logs that Exchange produces will contain several different fields of information. This information includes:

- **Date-time** – The date and time at which the logged event has occurred.
- **Connector-ID** – This field will be left blank, as it is not used by POP3 or IMAP4.
- **Session ID** – Each SMTP session is assigned a GUID that Exchange uses as the session ID. Events that are a part of the same SMTP session all share the same GUID.
- **Sequence Number** – The sequence number starts at zero, and is incremented for each event that occurs within the session. New sessions restart at zero.
- **Local endpoint** – The IP address of the local endpoint.
- **Remote Endpoint** – The remote endpoint's IP address.
- **Data** – Specific information on the event that has occurred.
- **Context** – This field will be blank because it is not used for POP3 or IMAP4 logging.
- **Event** – The event is represented by a single character. Here is a breakdown of what those characters mean:

+	Connect
-	Disconnect
>	Send
<	Receive
*	Information

ActiveSync

Exchange ActiveSync is a Client Access Server component that allows Exchange mailboxes to be synchronized with mobile devices. In addition to synchronizing mailboxes, ActiveSync also allows for the synchronization of calendars, task lists, and contacts.

ActiveSync Mailbox Policies

ActiveSync policies are policies that you can apply to users or to groups of users. In Exchange 2007 there was one global ActiveSync policy. In Exchange 2010, you can create several different ActiveSync policies and assign them on an as needed basis.

In order for an ActiveSync policy to be effective, the mobile device must be compatible with the policy. Devices that fully support ActiveSync policies are said to be fully provisionable. Older versions of Windows Mobile may support some, but not all of the available ActiveSync policy settings. Non Windows devices are generally considered to be non provisionable.

Here is a summary of the individual settings that you can implement within an ActiveSync policy:

- **Allow Bluetooth** – This setting allows you to enable or disable Bluetooth connections on mobile devices. As an alternative, you can choose to allow Bluetooth to be used solely for hands-free use.
- **Allow Browser** – You can use this setting to enable or disable Internet Explorer. This setting does not have any impact on the use of third party browsers.
- **Allow Camera** – This setting enables or disables the device's camera.

- **Allow Consumer Mail** – You can use this setting to prevent users from being able to set up a personal POP3 or IMAP4 based e-mail account.
- **Allow Desktop Sync** – You can use this setting to either allow or disallow desktop synchronization via USB , infrared, or Bluetooth connection.
- **Allow HTML E-mail** – This setting controls whether Exchange mail is sent to the device in HTML or plain text format.
- **Allow Internet Sharing** – You can use this setting to either allow or prevent the use of the device as a modem for laptop computers.
- **AllowIrDa** – The AllowIrDa setting controls whether or not devices will be able to establish infrared connections.
- **Allow Non-Provisionable Devices** – Disabling this setting prevents users from synchronizing Exchange to any mobile device that does not fully support ActiveSync policies. This includes older Windows Mobile devices that are partially provisionable.
- **AllowPOPIMAPEmail** – This setting allows you to restrict whether or not the device can be used with a POP3 or IMAP4 mail account.
- **Allow Remote Desktop** – You can use this setting to allow or prevent Remote Desktop sessions from being established from the device.
- **Allow Simple Passwords** – If this setting is enabled, then users will be able to use simple passwords such as ABCD.
- **Allow S/MIME Software Certificates** – This setting controls whether or not S/MIME certificates can be stored on the device.
- **Allow Storage Card** – The Allow Storage Card setting gives you the option of preventing users from reading data from or writing data to external storage cards.
- **Allow Text Messaging** – You can use the Allow Text Messaging setting to prevent users from sending SMS text messages from their phone.
- **Allow Unsigned Applications** – You can use this setting to ensure that an application can only be installed on the device if it has been digitally signed.
- **Allow Wi-Fi** – The Allow Wi-Fi setting controls whether or not mobile device users can access wireless access points.
- **Alphanumeric Password Required** – If you enable this setting, then users will be required to use mobile device passwords that contain a mixture of numbers and letters.
- **Approved Application List** – You can use the Approved Application list to control the applications that users are allowed to run on mobile devices.
- **Attachments Enabled** – If the Attachments Enabled setting is enabled, then users will be allowed to download E-mail attachments to their mobile devices.
- **Device Encryption Enabled** – Enabling this setting allows (but does not require) data stored on mobile devices to be encrypted. Many mobile devices do not support this setting.
- **Password Enabled** – You can use this setting to require mobile users to use a password on their device.
- **Password Expiration** – This setting allows you to control how frequently users are required to change their passwords.

- **Password History** – This setting controls the number of passwords that are retained for each user. Passwords cannot be reused until they are purged from the password history.
- **Policy Refresh Interval** – You can use this setting to control how often mobile devices check for changes to the ActiveSync policy.
- **Maximum Attachment Size** – The Maximum Attachment Size setting is used to prevent users from downloading excessively large message attachments to mobile devices.
- **Maximum Calendar Age Filter** – This setting controls the maximum number of days' worth of calendar data that will be synchronized to mobile devices.
- **Maximum Failed Password Attempts** – You can use the Maximum Failed Password Attempts setting to control the number of invalid passwords that can be entered before a device is wiped.
- **Maximum Inactivity Time Lock** – If this setting is used then the device is locked when it has been inactive for the specified amount of time.
- **Minimum Password Length** – This setting establishes the minimum password length.
- **Maximum E-Mail Age Filter** – This setting specifies the number of days' worth of e-mail messages that will be synchronized with a mobile device.
- **Maximum HTML E-Mail Body Truncation Size** – You can use this setting to control the maximum size of HTML e-mail messages. Messages exceeding the specified number of KB are truncated.
- **Minimum Device Password Complex Characters** – This setting controls the minimum number of complex characters that must be included in a user's password. Complex characters are considered to be any character other than a letter.
- **Maximum E-Mail Body Truncation Size** – Messages exceeding the number of KB stated in this policy are truncated.
- **Password Recovery** – This setting provides protection against forgotten passwords. If enabled, it allows a user to send a recovery password to their mailbox. They can use the recovery password to gain access to the mobile device.
- **Require Device Encryption** – Enabling this setting causes data stored on supported devices to be encrypted.
- **Require Encrypted S/MIME Messages** – If this setting is enabled, then S/MIME messages must be encrypted.
- **Require Manual Synchronization While Roaming** – This setting prevents excessive roaming charges by requiring devices to be manually synchronized any time the user is roaming.
- **Require Storage Card Encryption** – If this setting is enabled, then any data stored on a device's storage card must be encrypted.
- **Unapproved InROM Application List** – You can use this setting to prevent users from running certain applications that are built into mobile devices.

If you are going to be using any of the ActiveSync policy settings, you are required to have the necessary Exchange Client Access License. In addition, the following policy settings require an Exchange Enterprise Client Access License:

- **Allow Bluetooth**
- **Allow Browser**
- **Allow Camera**
- **Allow Consumer Mail**
- **Allow Desktop Sync**
- **Allow Internet Sharing**
- **Allow IRDA**
- **Allow Remote Desktop**
- **Allow Text Messaging**
- **Allow Unsigned Applications**
- **Allow Unsigned Installation Packages**
- **Allow Wi-Fi**
- **Approved Application List**
- **Unapproved InROM Application List**

You can create an ActiveSync policy by navigating through the Exchange Management Console to Organization Configuration | Client Access, and clicking on the New Exchange ActiveSync Mailbox Policy link. When you do, Exchange will launch the New Exchange ActiveSync Mailbox Policy Wizard, shown in **Figure 6**.

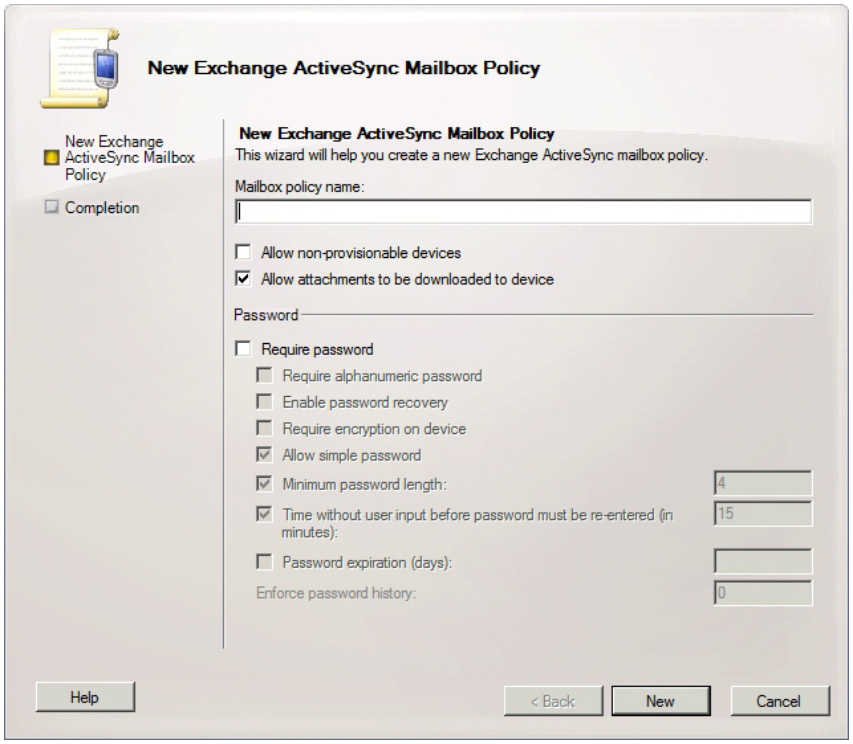

Figure 6: Creating a New Exchange ActiveSync Mailbox Policy

You can use the New Exchange ActiveSync Mailbox Policy Wizard to create a new ActiveSync policy.

As you can see in the figure, the wizard only provides for the most basic policy settings. To implement the other policy settings that have been discussed, you must create the policy and then right click on the newly created policy and choose the Properties command from the shortcut menu. The resulting properties sheet contains several tabs, each of which displays various policy settings.

Remote Wipe

Exchange Server 2010 client access servers include built-in wipe features that can blank a mobile device and return it to its factory defaults. Exchange 2010 allows for two different types of wipes – local and remote.

A local wipe is a wipe request that is triggered locally on a device. For instance, an ActiveSync policy may stipulate that a device will be wiped if a user enters their password incorrectly three times in a row.

A remote wipe is a wipe that is performed through the Exchange Control Panel if the device has been lost or stolen. When a remote wipe is performed, a confirmation of the wipe is sent back to Exchange.

When a device is wiped, any memory cards that are plugged into the device are also wiped. Although a wipe erases any data residing on a device and any attached memory cards, there are no guarantees as to whether or not the data can be recovered.

ActiveSync Reporting

The Client Access Server is capable of producing several different types of ActiveSync reports. These reports include:

- **Exchange ActiveSync Usage Report** – A general report that provides a detailed account of the volume of ActiveSync traffic that has passed through the server. This report also allows you to monitor ActiveSync traffic as it relates to specific categories of data (mail, calendar items, etc.).
- **Hits Report** – The Hits Report allows you to see the number of synchronization requests that are processed each hour, as well as the number of unique devices that are responsible for those requests.
- **HTTP Status Reports** – This is a general status report that you can use to gauge the health of the Client Access Server. It details the error codes that have been reported, as well as the frequency with which those errors are occurring.
- **Policy Compliance Report** – This report details the number of fully compliant, partially compliant, and non compliant devices that are making synchronization requests. A fully compliant device is one that fully complies with your ActiveSync policy. Partially compliant devices are generally older Windows Mobile devices that support some, but not all of the settings within the ActiveSync policy. Non compliant devices are devices to which the ActiveSync policy cannot be applied.
- **User Agent List** – The User Agent List displays each unique user who is using ActiveSync.

All of the ActiveSync logs are compiled by IIS, and are therefore technically IIS logs. To retrieve these logs, you should use the Export-ActiveSyncLog command.

Outlook Web App

Outlook Web App (OWA) is a Web based version of Outlook that is hosted on a Client Access Server. When you deploy a Client Access Server, Exchange automatically creates a virtual directory named OWA that is used by Outlook Web App.

Direct File Access

Direct File Access is a feature that allows users to use Outlook Web App as a mechanism for accessing files that are stored within file servers or in SharePoint document libraries. Private computer direct file access is enabled by default in Exchange 2010. The default behavior is to allow users to open files that are attached to e-mail messages.

Outlook Web App's behavior for opening files varies depending on the file type. Exchange maintains three separate lists of file extensions. These lists include:

- **Allow** – a list of the file types that users are allowed to open
- **Block** – a list of the file types that users are forbidden from opening
- **Force Save** – a list of the file types that users may open, but must first save to the local computer

If a file extension is accidentally included on multiple lists, then Allow takes precedence. The only time that the block list takes precedence is in situations in which a file type is also listed on the Force Save list, but not on the Allow list.

WebReady Document Viewing

WebReady Document Viewing is a feature which allows users to open certain types of files, such as Microsoft Office documents, in a Web browser. That way users can view message attachments even if they do not have the underlying application (such as Microsoft Office) installed.

You can configure WebReady Document Viewing through the Exchange Management Console by navigating through the console tree to Server Configuration | Client Access. Next, choose the OWA (Default Web Site) option from the work pane, and then click on the Properties link.

When the resulting properties sheet opens, you will notice that it contains otherwise identical tabs named Public Computer File Access and Private Computer File Access. These tabs correspond directly to the Outlook Web App logon options, which allow users to specify whether they are signing on from a public or from a private computer. To enable WebReady Document Viewing, select the Enable WebReady Document Viewing check box, as shown in **Figure 7**.

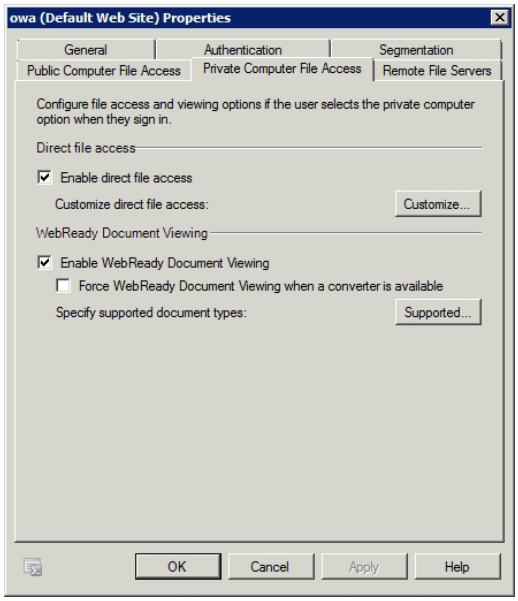

Figure 7: Configuring Private File Access Options

Exchange contains settings for making file shares available to users through OWA.

You can also enable WebReady Document Viewing from the Exchange Management Shell by using the following command:

```
Set-OwaVirtualDirectory -identity "owa (Default Web Site)"
-WebReadyDocumentViewingPublicComputersEnabled $true
```

Outlook Anywhere

Outlook Anywhere is a feature which encapsulates Remote Procedure Calls (RPC calls) inside of HTTP traffic. The reason for using this method is that is allows RPC traffic to traverse the firewall without having to open RPC specific ports.

Outlook Anywhere is a Client Access Server feature. It should be deployed on at least one Internet facing Client Access Server. In doing so, you will be able to use the same URL, namespace, and SSL certificate for Outlook Anywhere that you use for Outlook Web App or for Exchange ActiveSync.

Outlook Anywhere is an optional feature. If you do decide to enable it, you must do so on at least one Client Access Server. You can, however, enable Outlook Anywhere on multiple Client Access Servers if you so desire. Some possible reasons for doing so include improving performance and fault tolerance. From a performance standpoint, Microsoft recommends that you place at least one Outlook Anywhere enabled Client Access Server in each Active Directory site in which a mailbox server exists.

Outlook Anywhere can be used regardless of whether your Exchange organization contains all Exchange 2010 servers, or it contains a mixture of Exchange 2003, 2007, and 2010. In mixed environments, clients can use Outlook Anywhere to access their mailboxes regardless of which version of Exchange the server hosting their mailbox is running. The only caveat is that Exchange 2007 and Exchange 2003 servers must be manually configured to support Outlook Anywhere.

When you enable Outlook Anywhere on an Exchange 2010 Client Access Server, any user who has a mailbox on an Exchange 2010 server is automatically granted permission to use Outlook Anywhere.

The primary requirement for using Outlook Anywhere is that the Client Access Server must have a valid SSL certificate. Exchange 2010 includes a self signed certificate that can be used with Outlook Web App or with ActiveSync. However, the self signed certificate is not compatible with Outlook Anywhere.

As an alternative to deploying an SSL certificate directly onto the Client Access Server, some organizations make use of SSL offloading. SSL Offloading involves installing the SSL certificate onto the organization's firewall rather than installing it directly onto the Client Access Server. In doing so, traffic between the Internet and the organization's firewall is SSL encrypted, but traffic flowing between the firewall and the Client Access Server remains unencrypted. Although SSL offloading works and can sometimes improve performance, Microsoft recommends placing the SSL certificate on the Client Access Server so that all HTTP traffic can be encrypted from end to end.

The Autodiscover Service

The Autodiscover Service performs two primary tasks. These tasks include:

- The automatic configuration of user profile settings
- Providing Outlook 2007 and 2003 users with access to Exchange Server features

The Autodiscover service provides automatic configuration of user profiles for both Outlook clients and Windows Mobile clients.

For an Outlook client to benefit from the Autodiscover service, the user must be running Outlook 2007 or Outlook 2010. Likewise, the Autodiscover service is supported by Windows Mobile 6.1 and higher.

If an Outlook client or a Windows Mobile device has been joined to a domain, then the user's domain account is used for connectivity to the Autodiscover service. Otherwise, the user's e-mail address and password are used instead.

When an Outlook client or a Windows Mobile device connects to the Autodiscover service using the user's e-mail address and password, the Autodiscover service provides the following information to the client:

- The user's display name
- Connection settings for both internal and external connectivity
- Outlook Anywhere server settings
- The location of the user's mailbox
- The URLs associated with other Outlook features such as the Offline Address Book or Free / Busy information

The Autodiscover Service is automatically provisioned with the URLs used for resources such as the Availability Service and the Offline Address Book. However, if these services are to be externally accessible, then the Autodiscover service may need to be manually configured as the external URL is often different from the internal URL.

Making changes to the Autodiscover service involves using the Exchange Management Shell. The command that you will use in doing so varies depending on the URL that you are configuring. For example, if you need to configure the Autodiscover Service for Outlook Anywhere, you would use a command similar to this one:

```
Enable-OutlookAnywhere -Server CAS1 -ExternalHostname "mail.Domain1.
com" -ExternalAuthenticationMethod "Basic" -SSLOffloading:$False
```

This is a command that can be used to configure the Autodiscover Service for use with the Offline Address Book:

```
Set-OABVirtualDirectory -identity "CAS1\OAB (Default Web Site)"
-externalurl https://mail.Domain1.com/OAB -RequireSSL:$true
```

The command used with the Exchange Web Services is:

```
Set-WebServicesVirtualDirectory -identity "CAS1\EWS (Default Web Site)"
-externalurl https://mail.Domain1.com/EWS/Exchange.asmx -BasicAuthentication:$True
```

Federated Sharing

Although it is easy to think of an Exchange Server organization as a self contained structure, end users often need to collaborate with vendors, clients, and other non employees. Exchange Server 2010 offers a feature called federation which makes it possible for users to share contact information and free / busy information with others who are outside of the Active Directory forest.

Federated trusts are nothing new. They have existed in one form or another since the days of Windows NT. However, federated trusts between Exchange 2010 organizations work differently than what you might be used to.

In some of the previous versions of Windows, trusts were established directly between Active Directory forests. What some large organizations found, however, was that managing trusts could become quite messy due to the sheer number of forests that needed to be connected to each other.

In Exchange Server 2010, Microsoft has solved this problem by using the Microsoft Federation Gateway. The Microsoft Federation Gateway makes it possible to establish a trust between your Exchange 2010 organization and a cloud based federation server that is owned by Microsoft.

When a user is authenticated by the Active Directory, Microsoft's Federation Gateway issues the user a Security Assertion Markup Language (SAML) delegation token. This token is universal in nature, and it allows the user to be positively identified by any external Exchange 2010 organization that is connected to the Microsoft Federation Gateway.

Although the Microsoft Federation Gateway is a key component that is used in the federation process, it is not used automatically. After all, Microsoft does not have any reason to trust your Exchange organization. Likewise, your organization does not automatically trust the Microsoft Federation Gateway. Instead, you must manually establish a trust relationship between your organization and the Microsoft Federation Gateway.

You can create a federated trust by using either the GUI or by using the command line. If you choose to use the GUI, then you can create a federated trust by running the New Federation Trust Wizard. If you prefer to create a trust from the command line, then you will use the New-FederationTrust cmdlet.

In either case, Exchange will generate an application identifier (AppID) for your organization. The AppID is used by the Microsoft Federation Gateway to positively identify your Exchange organization. Microsoft also uses the AppID as a mechanism to prove that you actually own the domain name that your Exchange organization is using.

The reason why the AppID can be used to verify domain ownership is because of an underlying requirement. Any time that you want to establish a federated trust, you must create a TXT record in the DNS zone for each federated domain.

This requirement holds true for every accepted domain used by the Exchange organization. Every accepted domain must be added to the application identifier and must also have an associated TXT record. If you neglect to fulfill these requirements for an accepted domain, then users who have mailboxes within the domain will not be allowed to use federated sharing because Exchange will be unable to create the necessary account namespaces for those mailboxes.

In addition to the application identifier and DNS record requirements, organizations wishing to establish a federated trust must also have the necessary certificate in place.

Before federated trust can be established, the Exchange Server must be provisioned with an X.509 certificate. This certificate is different from the SSL certificate that Exchange uses to provide SSL encryption to Outlook Web App, Exchange ActiveSync and Outlook Anywhere.

The thing that makes the federation certificate different from the other SSL certificates that Exchange uses is that the certificate does not require a subject name or a subject alternate name.

The certificate that you use for federation must be issued by a trusted certification authority, and must be designated as an authentication certificate. Furthermore, the certificate must contain a subject key identifier (which most commercial certificate authorities provide). Finally, the certificate must use RSA as its signature algorithm, and it must have an exportable private key.

One additional thing that is worth mentioning about the X.509 certificates is that you do not have to provide a separate certificate for each individual Client Access Server. Instead, the certificate is treated as an organization level component and is replicated to the servers within the Exchange organization as is required.

Organization Relationships

Once an organization has been federated, they can exchange availability and other information with another federated organization through the use of an Organization Relationship.

An organization relationship is a one-to-one trust between two separate Exchange organizations. In order to establish an organization relationship, there are several requirements which must be met. These requirements include:

- At least one Exchange Server 2010 Client Access Server must exist in both organizations.
- Both organizations participating in the organization relationship must have been federated.
- A federation identifier must have been configured for both organizations.
- The organization identifier must include references to all accepted e-mail domains.

Once an organization relationship has been established, users in trusted organizations will be able to access free / busy information for the users in your organization. Normally, this information is made available automatically without you having to replicate the Global Address List to the trusted organization. However, Global Address List synchronization is required if users are running Outlook 2007.

If a user does not want to make their free / busy information available, they have the option of changing the underlying permissions from within Outlook. In Outlook 2010 this can be accomplished by going to Calendar Properties and selecting the Permissions tab. Next, select the default permission and set the permission level accordingly. Whatever permissions you choose to use will apply to both users in the local Exchange organization and users in trusted remote organizations.

Outlook calendar permissions are not the only mechanism for controlling the way in which a user's free / busy information is exposed to users in trusted organizations. Users actually have the ability to control which domains in trusted organizations have access to their free / busy information through the use of sharing policies.

Sharing policies are established on a per mailbox basis. It is up to the individual user to invite users in trusted organizations to share their free / busy information.

A sharing policy contains a pair of domain names (a local domain and a trusted federated domain) and a sharing action that is to be applied to the domain pair. The following sharing actions can be specified:

- Calendar sharing with free / busy information only.
- Calendar sharing with free / busy information, plus subject and location.
- Calendar sharing with free / busy information plus subject, location, and body.
- Contact sharing.
- Calendar sharing with free / busy information only. Contact Sharing.
- Calendar sharing with free / busy information, plus subject and location. Contact sharing.
- Calendar sharing with free busy information, plus subject, location, and body. Contact sharing.

Create a Federated Trust

A federated trust can be created by using either the Exchange Management Console or the Exchange Management Shell. To create a federated trust from the console, select the Organization Configuration container and then click on the New Federated Trust link. When you do, Exchange will launch the New Federated Trust wizard.

Go to the Certificate Thumbprint field and click Browse. Next, select the certificate that you want to use for the new federated trust. When you are done, click the New button. You should now be taken to the Completion page, where you can verify that the federated trust was successfully created. Click the Finish button to complete the process.

If you want to create a federated trust from the command line, the first thing that you will have to do is to get a list of the available certificates and their thumb prints. You can do so by entering the following command:

```
Get-ExchangeCertificate | Where {$_.IsSelfSigned -eq $false} | Format-List
```

After running this command, make note of the thumb print for the certificate that you want to use. It will consist of a long hexadecimal number. Once you have this number, you can create the federated trust by entering the following command:

```
New-FederationTrust –Name "Domain1 Federated Trust" –Thumbprint <thumb print number>
```

In the command shown above, you would replace <thumb print number> with the actual hexadecimal number that is associated with the thumb print.

When the federated trust is created, be sure to make note of the AppID, as you will need it during the configuration process.

Creating a TXT Record

As mentioned earlier, you must create a TXT record on your DNS server for each accepted domain that you include in a federated trust. This requirement helps to ensure that you actually own the domains for which you are creating the trust.

When you create the TXT record, you will do so through the DNS Manager console. Before you begin creating the TXT records, be sure that you have the AppID for your federated trust handy.

To create a TXT record, complete these steps:

1. Open the DNS Manager and select the Forward Lookup Zones container.
2. Select the forward lookup zone in which you want to create the TXT record.
3. Right click on the forward lookup zone, and select the.
4. Other New Records command from the resulting shortcut menu.

At this point, you should be taken to a dialog box that will allow you to create a new DNS record. Choose the Text (TXT) option from the Resource Record Type field, as shown in **Figure 8**, and then click on Create Record.

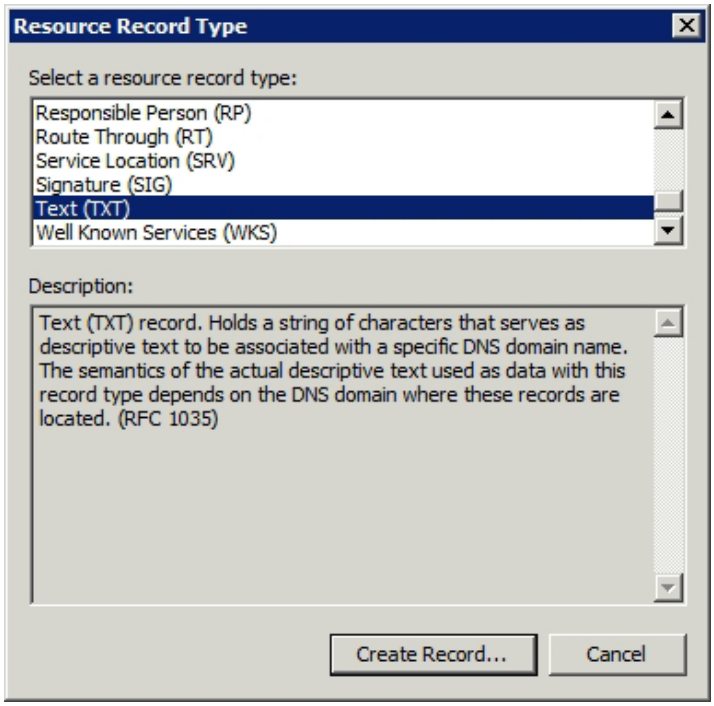

Figure 8: Creating a Resource Record

Choose the Text (TXT) option and click Create Record.

You will now be prompted to enter some details for the record that you are creating. You should leave the record name blank, as the record name needs to match the domain name. Likewise, the Fully Qualified Domain Name (FQDN) field should be populated automatically. This field is read only, so you don't have to do anything. You must populate the Text field with the AppID that was assigned to you when you created the federated trust. You must enter AppID= followed by the actual AppID number. For example, the AppID text might look something like this:

```
AppID=000000005112B67B
```

Click OK to complete the record creation process.

Creating an Organization Relationship

After you have created a federated trust and the necessary TXT records on your DNS server, you can create an organization relationship. Organization relationships can be created using either the Exchange Management Console or the Exchange Management Shell.

To create an organization relationship using the Exchange Management Console, select the Organization Configuration container and then click on the New Organization Relationship link. When you do, Exchange will launch the New Organization Relationship wizard, shown in **Figure 9.**

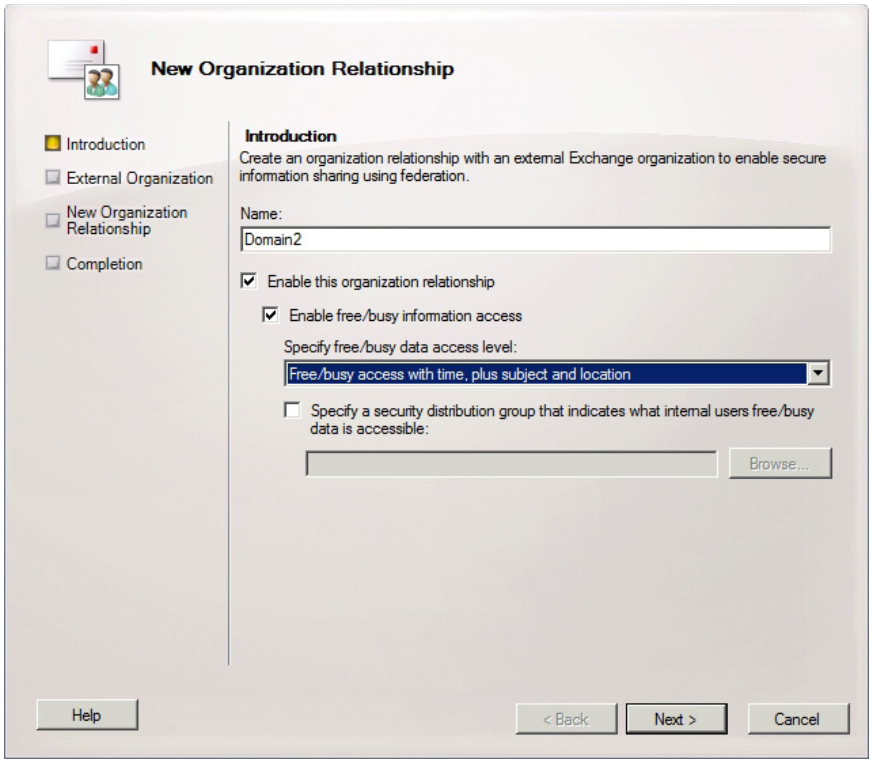

Figure 9: Configuring a New Organization Relationship

You must provide a name for the new organization relationship.

Begin populating the wizard's Introduction page by providing a name for the organization relationship that you are creating. Next, select the check box to enable the organization relationship. Assuming that you want to provide the trusted organization with free / busy information, you should select the Enable Free / Busy Information Access check box.

If you choose to allow Free / Busy Information access, you must choose the level of the information that you want to allow. Your choices include:

- No Free / Busy Access
- Free / Busy With Time Only
- Free / Busy With Time, Plus Subject and Location

At this point, you must specify which user's Free / Busy information will be accessible. Although you can make everyone's Free / Busy information accessible, you don't have to. Instead, you can specify a distribution group containing members whose Free / Busy information should be used.

Upon completion of this page, you will be taken to the External Organization page. This is where you provide information about the organization that you want to create a relationship with. Begin by clicking the Automatically Discover Configuration Information button, so that Exchange can use the Autodiscover service to find the trusted organization's configuration information.

Next, you must populate the Specify a Federated Domain of the External Exchange Organization field with the federated domain name of the trusted Exchange organization. For example, you might enter Domain1.com.

If you want to manually configure the organization relationship, then you can do so by clicking on the Manually Enter the Configuration Information button.

As with the automatic configuration, you are required to enter the federated domain name of the external Exchange organization. Additionally, you will have to enter the Application URI of the external Exchange organization as well as the URL used by the remote Exchange organization's Autodiscover service.

Just as you can use a wizard to create an organization relationship, so too can you use the Exchange Management Shell. Doing so involves using the New-OrganizationRelationship command. In conjunction with this command, you must supply a name for the organization relationship, the external domain names that you want to form a relationship with, and the level of free / busy access that you want to share. At its simplest, the resulting command looks something like this:

```
New-OrganizationRelationship –Name "Domain1" –DomainNames "Domain1.
com" –FreeBusyAccessEnabled $True –FreeBusyAccessLevel LimitedDetails
```

Create A Sharing Policy

As was explained earlier, sharing policies allow you to control how users are able to share calendars and contacts. You can create a sharing policy either through the Exchange Management Console or through the Exchange Management Shell.

To create a sharing policy using the Exchange Management Console, navigate through the console tree to Organization Configuration | Mailbox, and then click on the New Sharing Policy link, located in the Actions pane.

When the New Sharing Policy Wizard appears, begin the configuration process by providing a name for the sharing policy that you are creating, as shown in **Figure 10**. Next, click the Add button to specify the domains to which the sharing policy applies. If you do not want to restrict the sharing policy to a domain subset, simply use the asterisks to indicate that the policy should apply to all domains.

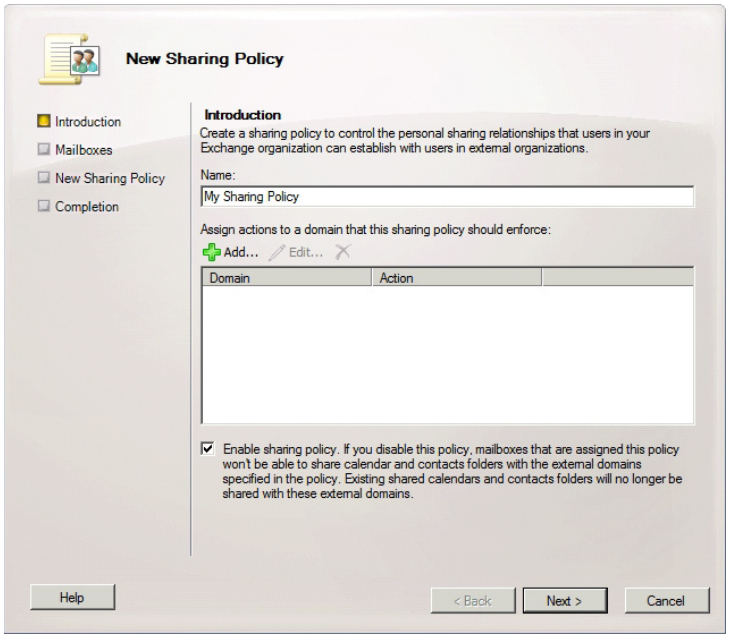

Figure 10: Creating a New Sharing Policy

You must assign a name to a new sharing policy.

While you are on this page, you will also have to specify the level of sharing that you want to provide. There is also a check box that you should select to enable the sharing policy.

On the following screen, click Add and then specify the mailboxes to which the policy should be applied. Finally, click New followed by Finish to complete the sharing policy.

Just as you can create a sharing policy from the Exchange Management Console, you can also create a policy from the command line by using the New-SharingPolicy command. In doing so, you must provide a name for the policy, the domains to which the policy should apply, and the level of Free / Busy access that you want to provide. Here is an example of what such a command might look like:

```
New-SharingPolicy "Domain1" -Domains Domain1.com: CalendarSharingFreeBusySimple
```

Once you have created a sharing policy, you can add mailboxes to it by using the Set-Mailbox command. For example, you might use a command like this:

```
Set-Mailbox -Identity JohnDoe -SharingPolicy "Domain1"
```

Of course, assigning sharing policies individually usually isn't practical. As such, it may be more effective to combine the Set-Mailbox command with the Get-Mailbox command in a way that allows you to apply the policy to multiple users at once. For example, if you wanted to apply the Domain1 policy to everyone in the IT department, you could do so by entering this command:

```
Get-Mailbox -Filter {Department -eq "IT"} | Set-Mailbox -SharingPolicy "Domain1"
```

Although these commands usually represent the most effective method for assigning a sharing policy to a mailbox, you can also assign sharing policies through the Exchange Management Console. To do so, navigate through the console tree to Recipient Configuration | Mailbox. Choose the mailbox that you want to manage, and then click on the Properties link. You should now see the mailbox's Properties sheet.

Go to the Mailbox Settings tab, select the Federated Sharing option and click Properties. Next, click the Browse button and then select the sharing policy that you want to assign. Click OK followed by Apply to complete the process.

Domain 4: Configuring Message Transport

The Hub Transport

Much of the 70-662 exam focuses on hub transport issues such as transport rules and e-mail address policies. This section covers exam material related to the hub transport server.

Accepted Domains

An accepted domain is a domain for which Exchange can receive or relay mail. All accepted domains must be explicitly defined within Exchange as an organizational level setting. Although edge transport servers must also be aware of the organization's accepted domains, the list of accepted domains is replicated from the Active Directory to the Edge Transport Server. Therefore, it is only necessary to configure the accepted domains at the organization level. You do not have to manually configure the accepted domain list on edge transport servers (although you can).

You can define an accepted domain by using either the Exchange Management Console or the Exchange Management Shell. To create an accepted domain using the Exchange Management Console, navigate through the console tree to Organization Configuration | Hub Transport, and then select the Accepted Domains tab located in the work pane. Next, click on the New Accepted Domain link to launch the New Accepted Domain Wizard.

When the wizard begins, you will be prompted to enter a friendly name for the new accepted domain as well as the actual domain name of the accepted domain. For example, you might use Domain1 as a friendly name and Domain1.com as the Accepted Domain name, as shown in **Figure 11**.

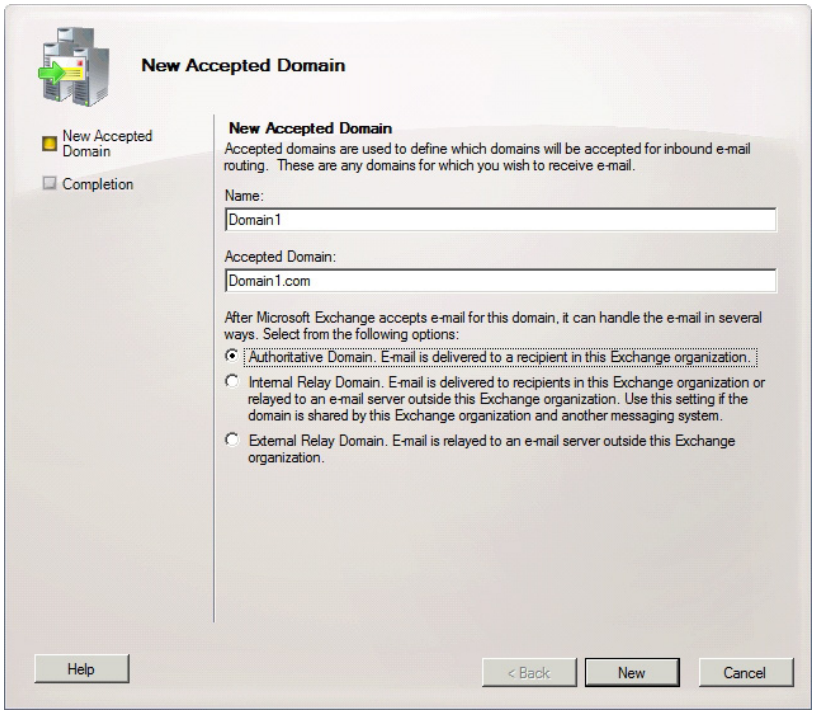

Figure 11: Configuring a New Accepted Domain

You must provide a friendly name and a fully qualified domain name for the new accepted domain.

At this point, you will be prompted to specify the type of accepted domain that you are creating. You have three choices:

1. **Authoritative Domain** – Used for recipients who have a mailbox on one of your mailbox servers.
2. **Internal Relay Domain** – An internal relay domain is used when you need to specify a domain that is outside of the Exchange organization, but that still belongs to the company.
3. **External Relay Domain** – An external relay domain is used when you need to relay messages to a domain that is not under your company's control.

Just as you can use the Exchange Management Console to create an accepted domain, you can also define an accepted domain by using the Exchange Management Shell. The command used in doing so is New-AcceptedDomain. When you use this command, you must provide a friendly name for the accepted domain, the domain name, and the domain type. Here is an example of how this command is used:

```
New-AcceptedDomain –Name "Domain1" –DomainName Domain1.com –DomainType InternalRelay
```

Authoritative Domains

When you configure the first hub transport server in your organization, that server is automatically provisioned with a default accepted domain. This default accepted domain is said to be authoritative for the organization. An authoritative domain is any domain for which Exchange hosts mailboxes.

Relay Domains

Relay domains, which are sometimes referred to as remote domains, are used far less frequently than authoritative domains. Relay domains are used when mail needs to be routed to recipients who do not have mailboxes on a mailbox server within the Exchange organization.

There are two types of relay domains – internal and external. Internal relay domains are most commonly used when an organization has multiple mail systems, but needs to send and receive mail from a common address space. Suppose, for instance, that an organization named Domain1 purchases a company named Domain2. Domain1 already owns the Domain1.com domain and uses that domain name for both internal and external mail.

Since Domain 2 was previously a separate company, they have their own mail servers in a completely separate Active Directory forest. The Domain 2 servers are configured to send and receive mail for the Domain2.com domain, but Domain1 needs mail sent to and from the Domain2 mail servers to be branded with the Domain1.com domain name even though Domain2 is using an entirely set of mail servers.

The first step in making this happen is for Domain1 to classify Domain2.com as an internal relay domain (you will also have to create a send connector). After doing so, suppose that someone sends an e-mail message to JohnDoe@Domain1.com. Let's also pretend that there is no user named JohnDoe in Domain1.com, but there is such a user at Domain2.com. Rather than returning a non-delivery report, the Domain1 mail server forwards the message to the internal relay domain Domain2.com, where the message is eventually delivered to the user JohnDoe.

External relay domains work similarly to internal relay domains, but are used in situations in which the organization does not own the relay domain. This type of configuration is sometimes used as a filtering mechanism. For example, an organization's MX record might be pointed to a server in a relay domain rather than to the organization's actual e-mail domain. In doing so, the external relay domain receives the messages and then performs any necessary filtering. The messages are then forwarded to their ultimate destination through a send connector.

E-Mail Address Policies

Before a recipient can send or receive any messages, they must be assigned an e-mail address. This is where the e-mail address policy comes into play. The e-mail address policy is responsible for generating a recipient's e-mail address, as well as an optional secondary address.

Exchange Server 2010 comes equipped with a default e-mail address policy. This default policy generates an e-mail address for each mail enabled user by combining the user's alias with the organization's default accepted domain. Although the default e-mail address policy is often sufficient, you can customize it or create additional policies.

You can create an e-mail address policy by using either the Exchange Management console or the Exchange Management Shell. To create an e-mail address policy using the console, navigate through the console tree to Organization Configuration | Hub Transport. Upon doing so, choose the New E-Mail Address Policy option. This will cause the console to launch the New E-Mail Address Policy wizard.

The first thing that you must do is provide a name for the policy that you are creating, as shown in **Figure 12**. This is the name that you will see any time you attempt to apply the policy to a mail enabled object or make a modification to an E-Mail Address Policy.

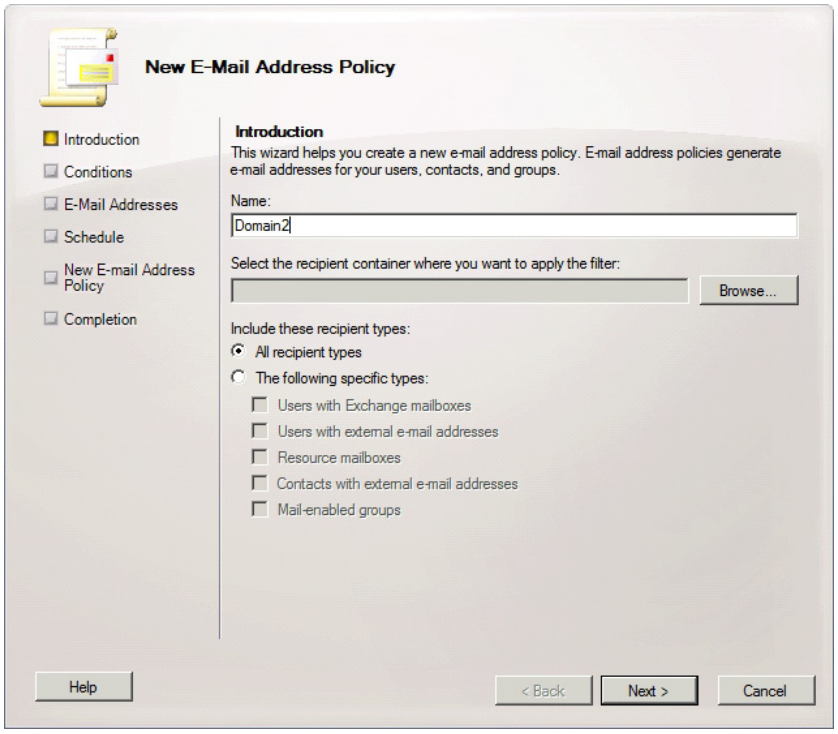

Figure 12: Assigning New E-Mail Address Policies

Exchange provides a wizard that you can use to create a new E-Mail Address Policy.

The next thing that you must do is to select a recipient container where you want the policy to apply. After doing so, you must choose the types of recipients that the new policy should apply to. You can choose to include all recipient types, or you can specify any combination of the following types of recipients:

- **Users with Exchange Mailboxes** – This includes users with Exchange 2010, 2007 or 2003 mailboxes.
- **Users with External E-mail Addresses** – This option pertains to users who have an Active Directory account within the organization, but who use an external e-mail address rather than an Exchange e-mail address. Choosing this option allows such users to be included in the Global Address List and in distribution lists.
- **Resource Mailboxes** – Resource mailboxes include equipment mailboxes and room mailboxes.
- **Contacts with External E-Mail Addresses** – Contacts With External E-Mail Addresses are similar to users with external e-mail addresses. The difference is that they do not have an Active Directory user account, and cannot log into the domain. Contacts with External E-Mail Addresses are created solely for the purpose of allowing external recipients to appear in the Global Address List.
- **Mail Enabled Groups** – This option allows the E-Mail Address Policy to be applied to distribution groups.

Once you have entered these options, click Next to go to the Conditions page. The Conditions page contains a series of check boxes that you can use to stipulate the conditions under which the address policy will be applied. You can set the following conditions:

- Recipient is in a State or Providence
- Recipient is in a Department
- Recipient is in a Company
- Custom Attribute Equals Value

On the following page you must specify the e-mail address that will be used for those users to whom the e-mail address policy applies. There are several different addresses that you can choose from. In almost every case you will use an SMTP address.

Microsoft gives you two options for creating SMTP addresses. You can create a precanned address, which is an SMTP address that adheres to a common format, or you can create a custom address. If you choose to create a custom address, there are several variables that you can use when forming the address. These variables tell Exchange to insert various pieces of information into the address. The variables that are available to you include:

- **%G** – Given name (first name)
- **%I** – Middle Initial
- **%S** – Surname (last name)
- **%D** – Display Name
- **%M** – Mailbox Alias
- **%XS** – The first X letters of the Surname. For instance, if you wanted to use the first letter of the user's last name then you would replace the X with the number 1
- **%XG** – The first X characters of the user's first name

Once you have told Exchange what format you want to use for e-mail addresses, you must click Next. When you do, you will be taken to the wizard's Schedule page. This is where you tell Exchange when the policy will go into effect. Your choices include:

- **Do Not Apply** – The policy is created, but is not applied to user mailboxes.
- **Immediately** – The policy is applied immediately.
- **At the Following Time** – Use this option to schedule the application of the newly created policy. If you use this option there is also an option to abort the process if it is still running after an excessive period of time. By default, scheduled policy creation is aborted if the process has not completed within 8 hours.

Once you have determined when the policy should be created, then click New to create the policy. When the process completes, click Finish to close the wizard.

Just as you can use the Exchange Management Console to create an e-mail address policy, so too can you use the Exchange Management Shell. To do so, you must use the New-EmailAddressPolicy cmdlet. Here is an example of how this cmdlet is used:

```
New-EmailAddressPolicy –Name "My Policy" –IncludeRecipients
MailboxUsers –EnabledEmailAddressTemplates "SMTP:%S@Domain1.com"
```

Transport Rules

Most organizations do not have the luxury of allowing e-mail messages to flow freely. Various laws and regulatory requirements typically mean that organizations must perform some filtering on messages entering or leaving the organization. For example, an organization may need to filter out messages containing inappropriate content or they may need to append a disclaimer to every message that is being sent. These tasks and more can be performed through the use of transport rules.

In an Exchange 2010 environment, every message passes through at least one hub transport server. Because of this, it is possible to use transport rules to filter messages while they are in transit.

Transport rules are processed by a Transport Rules Agent. Similarly, edge transport servers can also process transport rules via an Edge Rules Agent.

Although transport rules are an Exchange Server feature, the rules themselves are stored in the Active Directory. That way, they can be applied to every hub transport server in the entire Exchange Server organization without the administrator having to create and maintain a separate set of rules for each hub transport server.

Transport rules are made up of three components. These components include:

- **Conditions** – A condition is a situation that causes a rule to be applied. A condition is made up of predicates, which dictate the part of the message that must be examined in order to evaluate whether or not the rule should be applied. For example, a predicate might examine who a message is from or who it was sent to.
- **Exceptions** – Often times it would be inappropriate to apply a transport rule consistently across the entire organization. Therefore, Exchange allows you to create rule exceptions. For example, suppose that you created a rule that is designed to check every outbound message for a certain phrase and then forward a copy of any message containing that phrase to the HR department. Well, it would be silly to forward such a message to the HR department if the message was already being sent to or from the HR department. Therefore, you might create an exception that keeps the rule from being applied if the message is being sent to or from the HR department.
- **Actions** – An action is what happens when the rule is applied. In my previous example, the action was the forwarding of the message to the HR department.

Hub transport rules can be created through either the Exchange Management Console or through the Exchange Management Shell. To create a hub transport rule through the Exchange Management Console, navigate through the console tree to Organization Configuration | Hub Transport. Now, go to the Transport Rules tab located in the result pane and click on the New Transport Rule link. This will cause Windows to launch the New Transport Rule wizard, which is shown in **Figure 13**.

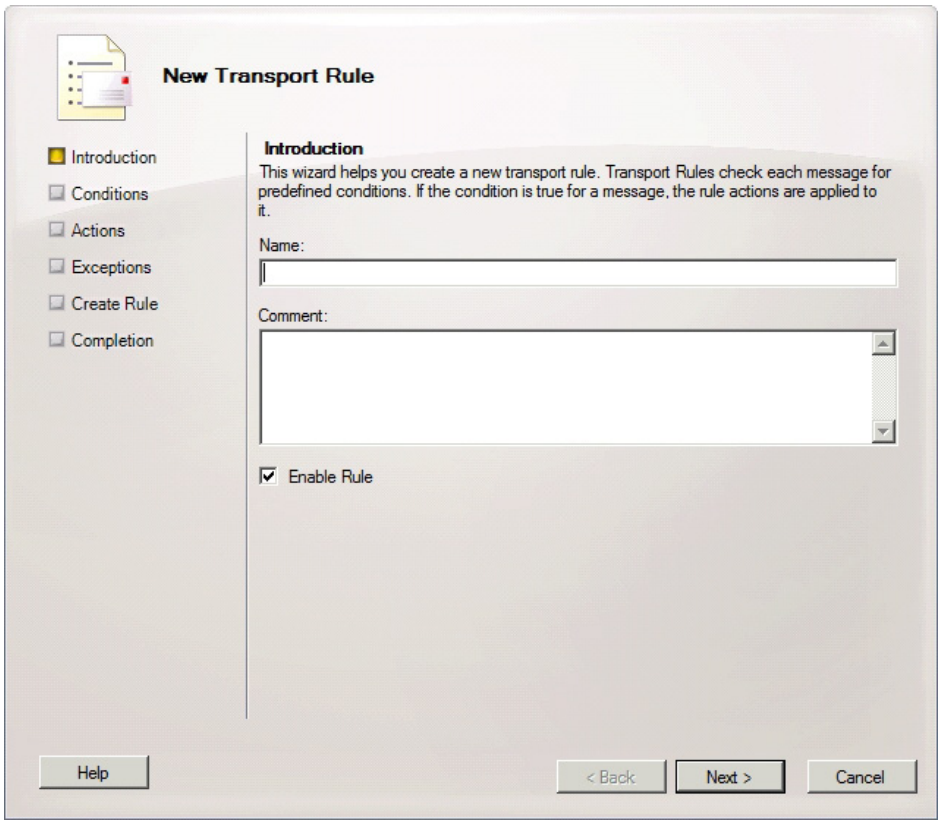

Figure 13: Creating a New Transport Rule

Transport rules consist of conditions, actions, and exceptions.

On the wizard's initial screen, you must enter a name for the rule, as well as an optional comment. Generally you should try to make the name and comment as descriptive as possible since you could eventually accumulate numerous rules and will need to be able to tell the rules apart. This screen also contains an Enable Rule check box which is used to control whether or not the rule is enabled after it is created. Hub transport rules are enabled by default.

The following screen asks you to set up the conditions for the rule. Usually you would select a condition from the Conditions dialog box. When you do, one or more words within the condition will be displayed in blue. Click on the blue words and then replace them with the values that you want the condition to look for.

Keep in mind that specifying a condition is not an absolute requirement. If you neglect to specify a condition, then the rule that you are creating will be applied to all messages.

The next screen that you will encounter asks you to define the actions for the rule. Begin the process by using the Select Actions box to choose the actions that you want to perform. After doing so, click on any words that are displayed in blue and edit them so that they reflect the actual values that you want the rule to use.

After you complete the process of defining the rule's actions, you are taken to the Exceptions screen. You do not have to create exceptions for every rule. If you want to create an exception, though, you can do so here by selecting the exception from the drop-down list, and then clicking on and editing any words that are displayed in blue.

At this point, you will be taken to a summary page that displays all of the information for the rule that you are about to create. Assuming that everything looks good, click New to create the rule. When the rule has been created, click Finish to close the wizard.

Although you can create transport rules from the Exchange Management Shell, it is usually much easier to create them from the console. The commands used to create transport lines can be very long and convoluted. This isn't always the case, though. Here is a simple example of how you can use the New-TransportRule command to create a transport rule:

```
New-TransportRule –Name "My Rule" –FromScope NotInOrganization –SentTo
"PR" –PrependSubject "Public Relations:"
```

The command above uses a condition and an action, but not an exception. The condition is that the message must come in from outside of the organization and be sent to someone in the PR group. The action is that the words Public Relations: will be added to the message's subject line.

Disclaimers

One of the most commonly used types of transport rules are disclaimers. Many organizations require that a legal disclaimer be appended to the bottom of all outbound e-mail messages. Sometimes, however, the disclaimer can be a uniform signature rather than a typical legal disclaimer.

The main thing that you need to know about disclaimers is that the disclaimer itself can be made up of either normal text or HTML code, as shown in **Figure 14**. HTML-based disclaimers can even contain images.

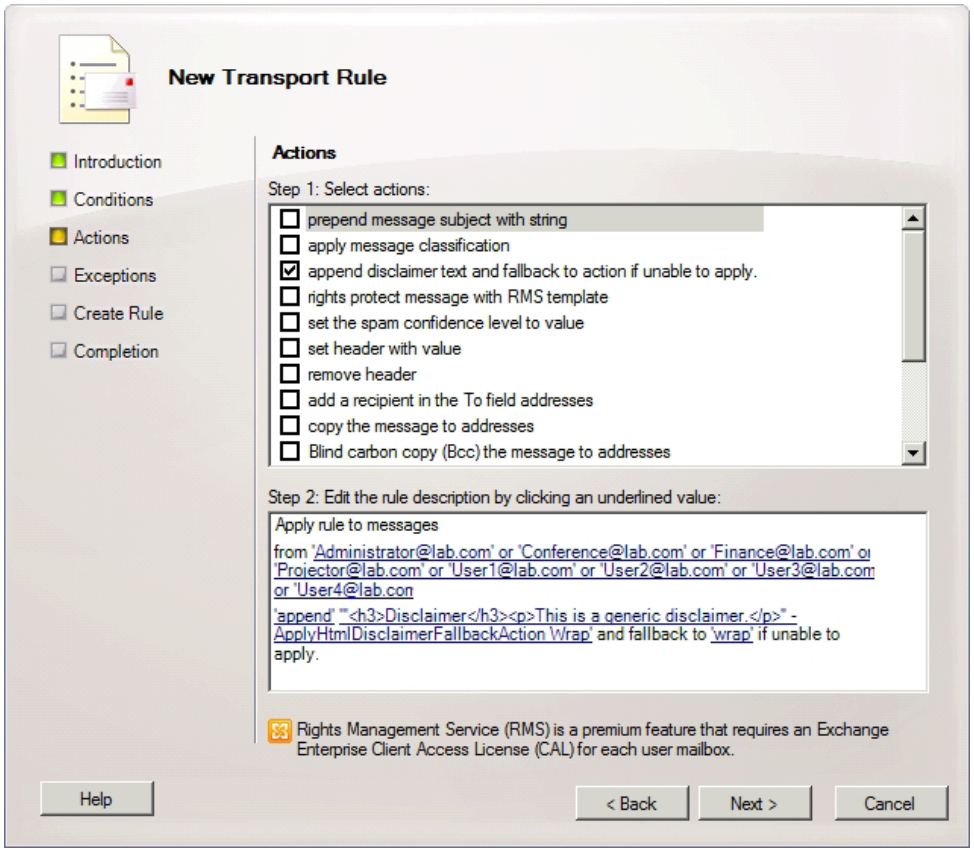

Figure 14: Setting Actions for the Transport Rule

A disclaimer can contain HTML code.

Since a disclaimer is really nothing more than a transport rule, you can create a disclaimer using exactly the same method as you would use for creating any other hub transport rule. The same can also be said for creating a disclaimer from the command line. The only thing that can sometimes be a little bit tricky is embedding HTML code within the command syntax. Here is an example of what such a command might look like:

```
New-TransportRule -Name ExternalDisclaimer -Enabled $true
-SentToScope 'NotInOrganization' -ApplyHtmlDisclaimerLocation
'Append' -ApplyHtmlDisclaimerText "<h3>Disclaimer</h3><p>This is a
generic disclaimer.</p>" -ApplyHtmlDisclaimerFallbackAction Wrap
```

Moderated Transport

Almost every company that is using Exchange 2010 has a distribution list set up that allows messages to be sent to every employee in the organization. Similar distribution groups may also exist that allow messages to be sent to all managers or to all of the employees within a certain department. Needless to say, you probably don't want just anyone sending messages to large distribution groups.

Microsoft's solution to this problem was to create Exchange 2010's moderated transport feature. The moderated transport, which is based on the Exchange 2010 approval framework, allows certain users to act as moderators for specific mailboxes (or distribution groups). The moderators must approve messages before they will be delivered to moderated locations.

There are four main components used in conjunction with the moderated transport. These components include:

- **The Categorizer** – The categorizer's job is to detect messages sent to moderated recipients, intercept the message, and send it to the arbitration mailbox (the moderator's mailbox).
- **Store Driver** – The store driver is an information store level component that checks to see whether or not messages sent to a moderated mailbox have been approved for delivery.
- **Information Assistant** – Once a message that has been sent to a moderated mailbox has been approved, the Information Assistant moves the message to the submission queue. If the moderator has rejected the message then the Information Assistant deletes the message.
- **Arbitration Mailbox** – The Arbitration Mailbox is the location in which messages sent to moderated mailboxes are initially delivered while they await approval.

You can designate a mailbox as being moderated by using either the Exchange Management Shell or the Exchange Management Console. To designate a mailbox for moderation using the Exchange Management Console, go to the Recipient Configuration container and select the mailbox or the distribution group that you want to moderate. Click on the Properties link found in the Actions pane. This will cause the console to open the mailbox or distribution group's properties sheet.

At this point, you should go to the properties sheet's Mail Flow Settings tab. Choose the Message Moderation option, and then click on Properties. When the Message Moderation dialog box is displayed, select the Messages Sent to This Group Have To Be Approved By A Moderator check box.

Next, click the Add button and select the user who you want to act as a moderator for the mailbox. You can also use the Add button within the Specify Senders Who Don't Require Message Approval section to create a list of users who are allowed to bypass the moderation process and send messages directly to the moderated mailbox. Finally, there is a check box that you can select if you want to notify the senders in your organization if the messages that they send to a moderated mailbox are not approved by the moderator.

Just as you can set up moderation through the Exchange Management Console, you can also use the Exchange Management Shell. To do so, you must use the Set-Mailbox or the Set-DistributionGroup command. Here is an example of such a command:

```
Set-DistributionGroup "Managers" –ModerationEnabled $True –ModeratedBy
Brien@Domain1.com –ByPassModerationFromSendersOrMembers "IT" –
SendModerationNotifications Internal
```

The command shown above sets up moderation on a distribution group named Managers. This command enables moderation and then designates Brien@Domain1.com as the moderator. Finally, members of the IT department are allowed to bypass the moderation process and moderation notifications are enabled for internal users.

Information Rights Management

One of the big problems with e-mail has always been the potential for data leakage. E-mail provides users with an easy mechanism for sending sensitive information to people outside of the company. The problem is that you never know what the recipient is going to do with the information once they get it. For example, the recipient might forward sensitive information that was intended to have been confidential.

This is where Information Rights Management (IRM) comes into play. Information Rights Management can prevent a message recipient from forwarding, printing, faxing, saving, or cutting and pasting a message's contents. Likewise, IRM can also protect message attachments in the same way, and it is even possible to set an expiration period for a message so that the recipient can no longer view the message after a certain length of time.

The Rights Management Service is not an Exchange service, but rather a Windows service. Rights management is applied through the use of templates. Exchange Server 2010 ships with a rights management template called Do Not Forward. When this template is applied to a message, the message is encrypted. Only the message's recipient has the rights to decrypt the message. Furthermore, the recipient is forbidden from forwarding the message to anyone else, printing the message, or copying the message.

Normally, the Do Not Forward template must be manually applied to messages by users through either Outlook 2010 or Outlook Web App. However, the template can be applied automatically either through Outlook rules or through hub transport rules. Outlook has supported the Rights Management Service since Outlook 2003. However, Outlook Protection Rules are new to Outlook 2010.

Implementing the Rights Management Service requires the deployment of an AD RMS cluster. Specifically, this means that you must set up a Windows Server that can act as an RMS server. Specifically, the RMS server must adhere to the following requirements:

- The server must be running Windows Server 2008 R2 or Windows Server 2008 with SP2 and hotfix 973247.
- When you deploy the RMS server, you must create a Service Connection Point (SCP) that Exchange can latch onto.
- Exchange Servers in your organization must have read and execute permissions for the RMS server's certification pipeline. By default, the certification pipeline path is: \inetpub\wwwroot_wmcs\certification\ServerCertification.asmx.
- The Federated Delivery Mailbox (which is a special Exchange 2010 mailbox) must be added to the Super Users group on the RMS server. Otherwise the various RMS related agents that Exchange uses will not work.
- Microsoft refers to an AD RMS server as an AD RMS cluster. This is just a term that Microsoft uses, and it doesn't mean that you have to deploy a failover cluster in order to use RMS. Single server deployments are supported, but are discouraged because if an RMS server fails it can prevent users from being able to access RMS protected messages.

The Rights Management Service Agents

As mentioned earlier, the Rights Management Service is a Windows service, not an Exchange service. As such, Exchange must be able to communicate with the Rights Management Server, This communication is facilitated through the use of various agents. Except for the pre-licensing agent, all of the agents reside on the hub transport server, but can't be controlled in the same way that you manage other transport agents. The agents that Exchange uses include:

- **The Pre-Licensing Agent** – Rights Management protected messages are encrypted and the recipient must be able to decrypt the message before they will be able to read its contents. Normally, this would require the recipient to communicate with the AD RMS server. However, the pre-licensing agent provides the recipient with the necessary Rights Management Service license and allows the message to be decrypted on the fly. This makes it possible for the recipient to view RMS protected messages while working offline or while working in Outlook Web App.
- **RMS Decryption Agent** – As the name implies, this agent allows RMS protected messages to be decrypted.
- **Transport Rules Agent** – Transport rules almost always processes messages based on their contents. Being that RMS protected messages are encrypted, the hub transport server needs a way of decrypting the messages so that the messages can be evaluated by the transport rules. The Transport Rules Agent decrypts RMS protected messages flowing through the transport pipeline for the purpose of applying transport rules.
- **RMS Encryption Agent** – Sometimes regulatory requirements force organizations to encrypt certain types of messages. In these types of situations, organizations cannot rely on end users to RMS protect messages. Instead, transport rules may be used to evaluate the messages and see if they need to be encrypted. If a message does need to be encrypted, the encryption is performed by the RMS Encryption Agent.
- **Journal Report Decryption Agent** – If an organization uses journaling to archive messages, then the archives would be useless if they were full of unreadable, RMS protected messages. As such, the Journal Report Decryption Agent decrypts RMS messages that are attached to journal reports, and embeds a clear text version of the message alongside the RMS protected version.

It is possible to open RMS protected messages on mobile devices. However, there are several criteria that must be met. These criteria include:

- The device must be running Windows Mobile 6.0 or later.
- The mobile device must be activated while connected to a computer that is running Windows 7, Vista, or XP.
- The computer used to activate the mobile device must be a domain member, and must be able to communicate with the RMS server.
- The RMS server must be configured to enable certification of mobile devices.

Rights Protection (Using Transport Rules)

As mentioned earlier, many organizations are required by law to encrypt messages containing certain types of business information or personally identifiable information. Exchange 2010 makes it possible to use transport rules to determine if a message requires RMS protection (based on content), and to automatically apply RMS protection when necessary.

The technique used for RMS protecting messages through transport rules is virtually identical to the technique that you would use to create any other transport rule. To do so, you must create a condition, an action, and an optional set of exceptions.

In this case, the condition is the mechanism used to determine whether or not the message needs to be protected. You might, for instance, create a condition that looks at the message's subject line, the message's body, or even the message's classification.

Microsoft offers an action that is specifically designed to RMS protect messages. The action is called Rights Protect Messages With RMS Template. When you choose this action, you must simply tell Exchange that you want to use the Do Not Forward template.

It is worth noting that transport rules can only be used to protect messages if the Prelicensing Agent is enabled. You can enable prelicensing through the Exchange Management Shell by entering the following command:

```
Set-IRMConfiguration -PrelicensingEnabled $true
```

Edge Transport

An Edge Transport Server (which is sometimes referred to as an edge server) sits at the network perimeter and shields the Exchange Server organization from inbound internet traffic. The Edge Transport Server's primary job is to filter viruses and spam from inbound messages, while also obscuring backend Exchange Servers.

Install the Edge Transport Server Role

Although the procedure for installing an Edge Transport Server is similar to the procedure used for installing other Exchange Server roles, there are a few differences. These differences are due to the unique nature of the Edge Transport Server role.

As you know, Exchange is completely dependent on the Active Directory. Even so, an Edge Transport Server cannot be a domain member. Instead, a minimal amount of directory information is replicated from the Active Directory (through the Hub Transport Server) to the Edge Transport Server. This is Microsoft's way of protecting the Active Directory against Internet based attacks.

Because of the nature of the Edge Transport Server, the Edge Transport Server role cannot be combined with any other Exchange Server roles. Therefore, the Edge Transport Server role can only operate on a dedicated server (although the server can be physical or virtual).

There are several prerequisites that must be met before you can begin installing Exchange on an Edge Transport Server. These prerequisites vary depending on whether your server will be running Windows Server 2008 with SP2 or higher, or Windows Server 2008 R2.

For a server that is running Windows Server 2008 SP2, the prerequisites include:

- The Active Directory forest functional level must be set to at least Windows Server 2003.
- You must install Microsoft .NET 3.5 with SP1 or higher.
- You must install the Microsoft .NET 3.5 Family Update for Windows Vista and Windows Server 2008.
- You are required to install Windows Remote Management 2.0 (WinRM 2.0).
- You must also install version 2.0 of Windows PowerShell.

The easiest way to install these prerequisites is to open an elevated command prompt window, and then navigate to the Scripts folder on the Exchange 2010 DVD, and run the following command:

```
ServerManagerCmd -ip Exchange-Edge.xml –Restart
```

Windows Server 2008 R2 already has many of the necessary prerequisites in place. Therefore, the easiest way to prepare Windows Server 2008 R2 to act as an Edge Transport Server is to open PowerShell and run the following commands:

```
Import-Module ServerManager
Add-WindowsFeature NET-Framework,RSAT-ADDS,ADLDS -Restart
```

Once the necessary prerequisites have all been installed, you can begin installing the Edge Transport Role. To do so, insert the Exchange Server 2010 installation DVD and wait for the splash screen to appear. Make sure that Setup indicates that Steps 1 and 2 have been completed.

1. On the splash screen, choose Step 3: Choose Exchange Language Option. Choose to either install all of the languages from the language bundle or to install only the languages that are found on the DVD.
2. At this point, click on Step 4: Install Microsoft Exchange. This will cause Setup to display an introductory screen. Click Next to bypass this screen.
3. You will now be taken to the License Agreement page. Choose the I Accept the Terms in the License Agreement check box, and click Next.
4. Setup will now display the Error Reporting page. Decide whether or not you want to enable error reporting, and click Next.
5. Setup should now ask you what type of installation you want to perform. Choose the Custom Exchange Server Installation option, and click Next.
6. You will now be taken to the Server Role Selection screen. Choose the Edge Transport Server role and click Next.
7. You should now see the Customer Experience Improvement page. Choose the selection that is appropriate for your organization, and click Next.
8. Setup should now perform some readiness checks to make sure that Exchange is ready to be installed. Assuming that the server passes the readiness checks, click Install to begin the installation process. Otherwise, correct whatever issues are reported and retry the installation. When the installation process completes, click Finish.

EdgeSync

Because the computer on which the Edge Transport Server has been installed does not have access to the Active Directory, the Hub Transport Server must read Exchange related information from the Active Directory and send a minimal amount of information to the Edge Transport Server. This process is called an Edge Synchronization or EdgeSync.

The first step in creating an edge synchronization is to create an edge subscription. The subscription creates a file that is used by the EdgeSync process. This file sets up the security that is necessary for the Edge Transport Server to trust the Hub Transport Server.

The first step in creating an edge subscription is to create the subscription file. To do so, you must open the Exchange Management Shell on your Edge Transport Server, and run the following command:

```
New-EdgeSubscription -FileName "C:\EdgeSubscriptionInfo.xml"
```

The command shown above creates an edge subscription file named EdgeSubscriptionInfo.xml. This file must be moved from the Edge Transport Server to a location within the backend Exchange Server organization where it can be imported. The file should be moved – not copied. Leaving the edge subscription file on an Edge Transport Server poses a security risk.

At this point, you can use either the Exchange Management Console or the Exchange Management Shell to import the edge subscription file into the Hub Transport Server. To use the Exchange Management Console, go to the Hub Transport Server and then open the management console (you can't use the Exchange Management Console found on the Edge Transport Server for this procedure).

Navigate through the console tree to Organization Configuration | Hub Transport. Select the Edge Subscription tab, found in the result pane. Now, click on the New Edge Subscription link found in the Action pane. This will cause Exchange to launch the New Edge Subscription wizard.

When the wizard starts, there are three fields that you have to fill in. These fields include:

- **Active Directory Site** – You must select the Active Directory Site in which the Hub Transport Server that will be linked to the Edge Transport Server resides.
- **Subscription File** – You must click the Browse button and then select the subscription file that you want to use.
- **Automatically Create a Send Connector for this Edge Subscription** – Selecting this check box automatically creates a send connector that routes outbound mail through the Edge Transport Server on its way to the internet.

When you have finished entering this information, click New, followed by Finish to create the subscription.

If you prefer, you can use the Exchange Management Shell to create the edge subscription. The command for doing so is:

```
New-EdgeSubscription -FileName "C:\EdgeSubscriptionInfo.xml" -CreateInternetSendConnector
$true - CreateInboundSendConnector $true -Site "Default-First-Site-Name"
```

As you can see, the command shown above specifies the name of the edge subscription file that is to be used. It also stipulates that an Internet Send Connector and an Inbound Internet Send Connector will both be created. Finally, the subscription is going to be bound to the Active Directory site named Default-First-Site-Name.

Configure Edge Transport Settings

Like any other Exchange Server, an Edge Transport Server comes equipped with a copy of the Exchange Management Console and the Exchange Management Shell. However, the management tools found on an Edge Transport Server have been reduced to provide only the functionality that is required for managing the edge server. There are several configuration properties that you can edit on an Edge Transport Server. You can access these properties in the Exchange Management console by selecting the Edge Transport container and clicking Properties. The configurable properties are divided among several different tabs. Here is a summary of the options that are available:

The General Tab

- **Version** – A read-only field that displays the version of Exchange running on the server.
- **Edition** – A read-only field indicating whether the server is running Standard or Enterprise Edition.
- **Roles** – This field should indicate that the server is hosting the Edge Transport Server role.
- **Product ID** – This field should display the product ID, unless you have not yet entered a product key for Exchange.
- **Modified** – The Modified field tells you when the most recent configuration change occurred.

The System Settings Tab

- **Automatically Send Fatal Service Error Report to Microsoft** – You can use this option to enable error reporting for the server.

The Customer Feedback Options Tab

- This tab is used to opt in or out of the Customer Experience Improvement program.

The External DNS Lookups Tab

- **Use Network Card DNS Settings** – This option tells the server to use the DNS settings that are bound to its network cards.
- **Use These DNS Servers** – You can use this option to provide the server with a list of DNS servers to use.

The Internal DNS Lookups Tab

- **Use Network Card DNS Settings** – This option tells the server to use the DNS settings that are bound to its network cards.
- **Use These DNS Servers** – You can use this option to provide the server with a list of DNS servers to use.

The Limits Tab

- **Outbound Connection Failure Retry Interval** – This value (which defaults to 30 minutes) controls how frequently Exchange will attempt to reestablish a connection after a loss of connectivity.
- **Transient Failure Retry Interval (Seconds)** – You can use this option to specify the amount of time that should pass between transient connection attempts.
- **Transient Failure Retry Attempts** – This option controls the number of times that a server will attempt to reestablish connectivity to a remote server in a failure situation.
- **Maximum Time Since Submission (Days)** – This option controls how long a message can remain in queue before it times out.
- **Notify Sender When Message is Delayed More Than (Hours)** – You can use this option to notify a sender that their message has not yet been delivered.
- **Maximum Concurrent Outbound Connections Per Domain** – This option limits the maximum number of concurrent connections to a domain. The default value is 20, but the valid range spans from 1 to 2,147,483,647.

The Log Settings Tab

- **Enable Message Tracking Log** – Message tracking is enabled by default, but you can use this option to disable it.
- **Message Tracking Log Path** – This option shows the path in which message tracking logs are being stored.
- **Enable Connectivity Log** – Connectivity logging is disabled by default, but if you are having connectivity problems you can enable connectivity logging by using this setting.
- **Connectivity Log Path** – This option displays the path used by the connectivity logs.
- **Send Protocol Log Path** – This field displays the locations of the logs generated by the Send connector.
- **Receive Protocol Log Path** – This field shows the path used by the receive connector logs.

Cloning an Edge Transport Server

Organizations that make use of Edge Transport Servers often choose to deploy multiple Edge Transport Servers as a way of improving performance and reliability. In these types of situations, it is important for Edge Transport Servers to be configured in a consistent manner. The easiest way to achieve a consistent configuration is to clone the existing configuration, rather than configuring each Edge Transport Server individually.

Keep in mind that the configuration information for an Edge Transport Server is stored in the Active Directory, and then replicated to the Edge Transport Server through the edge subscription. Cloning allows each Edge Transport Server to use a common set of configuration information.

The first step in creating a cloned configuration is to use the ExportEdgeConfig.ps1 script (located in the C:\Program Files\Microsoft\Exchange Server\Scripts folder) to export the edge server's current configuration. To do so, copy the script from its current location to the root folder of your user profile on the source server. Next, enter the following command:

```
./ExportEdgeConfig -CloneConfigData:"C:\CloneConfigData.xml"
```

This command creates a file named CloneConfigData.xml, and places it in the server's root directory on C:. You must copy this file to the target server.

The next step in the process is to copy the ImportEdgeConfig.ps1 file from the C:\Program Files\Microsoft\Exchange Server\Scripts folder to the root folder of your user profile on the target server. After doing so, you must use the script to validate the CloneConfigData.xml file. The command shown below assumes that this file resides in the root directory on the target server:

```
./ImportEdgeConfig -CloneConfigData:"C:\CloneConfigData.xml" -IsImport
$false -CloneConfigAnswer:"C:\CloneConfigAnswer.xml"
```

When you run the command shown above, Exchange will create an answer file named C:\CloneConfigAnswer.xml. Since this file may contain server specific settings, you should take a moment to open the file and make sure that no modifications are required. Once you have verified the information in the answer file, you can import it into the new edge server by running the following command:

```
./ImportEdgeConfig -CloneConfigData:"C:\CloneConfigData.xml" -IsImport
$true -CloneConfigAnswer:"C:\CloneConfigAnswer.xml"
```

Message Routing

Message routing refers to the way that Exchange moves messages throughout the Exchange Server organization. Exchange routes messages through a series of connectors.

Mail Connectors

Exchange Server 2010 makes use of several different types of connectors. These include:

Send Connectors - A send connector is a logical gateway responsible for sending messages to the next hop. A send connector can connect to a Hub Transport Server, an Edge Transport Server, or to a remote mail system. Exchange does not automatically create any send connectors when you install the hub transport role, but it is still able to route messages internally based on Active Directory site information.

Each send connector is assigned a specific address space. The address space reflects the domains that the connector can send mail to. The following types of address spaces can be used:

- * An asterisk indicates that the send connector can be used to route messages to any domain.
- A specific domain Entering a specific domain name (such as Domain1.com) limits the send connector to sending mail only to that domain.
- *.domain An asterisk in front of a domain name (such as *.Domain1.com) indicates that the connector can be used to send mail to the listed domain and to all sub domains.
- -- This address space is only used on Edge Transport Servers. It means that the send connector can send messages to any of the organization's accepted domains, but nothing else.

Receive Connectors – A receive connector is a listener that listens for inbound messages. Exchange automatically creates receive connectors on all hub transport servers. However, these receive connectors will initially only facilitate internal mail flow.

When you deploy an Edge Transport Server, Exchange will create a receive connector that allows the Edge Transport Server to receive mail from the internet. The Hub Transport Servers will also be able to route internet mail once an edge subscription has been established. The subscription process creates a receive connector that allows a Hub Transport Server to receive mail from an Edge Transport Server.

The default receive connectors are often sufficient, but Exchange allows you to create additional receive connectors as your needs dictate. For example, if your organization is not going to be using an Edge Transport Server, then you may need to create a receive connector that will allow a Hub Transport Server to receive internet mail for your domain.

Any time that you create a custom receive connector, it is important to properly secure the connector. Otherwise, you could end up with an open relay. Spammers use open relays as a mechanism for making it appear that their messages have come from someone else's mail server. Having an open relay is usually enough to land an organization on a spam blacklist.

Foreign Connectors – Foreign connectors are seldom used in the real world. A foreign connector is only used when Exchange needs to route messages to a system that does not support the use of SMTP. Since almost all mail servers support SMTP, foreign mail connectors are used primarily with fax gateways.

Sites and Costs

Normally, Exchange Server 2010 routes messages according to costs. Every Active Directory site link is assigned a cost value. Exchange takes the cost values associated with the various site links into account when determining which route to use when routing messages through the Exchange organization. Even so, cost is not the only factor that is taken into account when determining message routing paths.

Exchange also considers connector state when choosing a routing path. A connector will not be considered unless it is enabled. However, the connector state may be overlooked if your organization contains Exchange 2007 transport servers.

Another factor that is considered is the link connector. For example, if a receive connector is directly connected to a send connector, then the send connector will be used regardless of cost. Linked connectors always take precedence over any other actors when calculating a routing path.

Another factor that Exchange takes into account is the address space that is assigned to a send connector. Exchange will always try to use the send connector that is the best match for the destination based on address space. More specific address spaces take precedence over vague address spaces. If multiple send connectors connect to the same address space, then Exchange will revert to using cost to determine which connector to use.

As Exchange performs routing calculations, it also examines the connector scope. Some connectors may be configured in such a way that they are only accessible from specific Active Directory sites.

Likewise, some connectors place restrictions on the maximum size of any message that traverses the connection. Exchange will always take a message's size into account when choosing a routing path.

Although proximity often goes hand-in-hand with aggregate cost, Exchange does take proximity into account when choosing a routing path. Exchange gives preference to local servers, followed by servers in the current Active Directory site. Exchange Servers in remote Active Directory sites are considered to be the furthest away, and are therefore given the lowest priority.

Occasionally, Exchange may encounter a situation in which it is unable to determine the best routing path based on cost or on any of the other factors mentioned here. When this occurs, Exchange begins making alphanumeric comparisons. It begins by comparing the name assigned to the Active Directory site. The path where the site nearest to the destination is the lowest in alphanumeric order is the one that Exchange chooses to use. Just as Exchange may make an alphanumeric comparison of Active Directory site names, it may also compare the names of the routing group connectors in the same manner.

Domain 5: Monitoring and Reporting

Mailbox Database Statistics

Any time that you are working on capacity planning or on planning for the use of mailbox quotas or archive mailboxes, you will most likely have to gather some information regarding how the mailboxes are currently being used. Fortunately, Exchange 2010 includes a cmdlet that you can use to gather the necessary statistics. The cmdlet is Get-MailboxStatistics. Its syntax is as follows:

```
Get-MailboxStatistics –Identity <mailbox name> -Archive –Domain
Controller <domain controller>
```

The only required parameter in the command above is the Identity switch, which you would typically follow with the name of the mailbox. The mailbox name can be specified as one of the following:

- GUID
- Distinguished Name (DN)
- Domain\Account
- User Principle Name (UPN)
- Legacy Exchange Distinguished Name (DN)
- SMTP Address
- Alias

The Alias is used most commonly, but any of objects listed above can be used.

The Archive switch is optional. If you use this switch, you are telling Exchange that you want to gather statistics for the specified user's archive mailbox. Archive mailboxes are new to Exchange 2010 and are a secondary mailbox that can be used to store a user's archives.

The Domain Controller switch is also optional. You can use this switch as a way of directing Exchange to use a specific domain controller while gathering mailbox statistics. If you decide to use the Domain Controller switch, you must append the name of the domain controller in Fully Qualified Domain Name (FQDN) format.

Database Status

Just as you can use the Get-MailboxStatistics cmdlet to return the statistics for an individual mailbox, you can also use it to return the statistics for all of the mailboxes in an entire mailbox database. In doing so, you will use a slightly different syntax. The syntax required for returning database level statistics is:

```
Get-MailboxStatistics –Database "mailbox database name"
```

As you can see in **Figure 15**, the only attribute that you are required to provide is the name of the mailbox database.

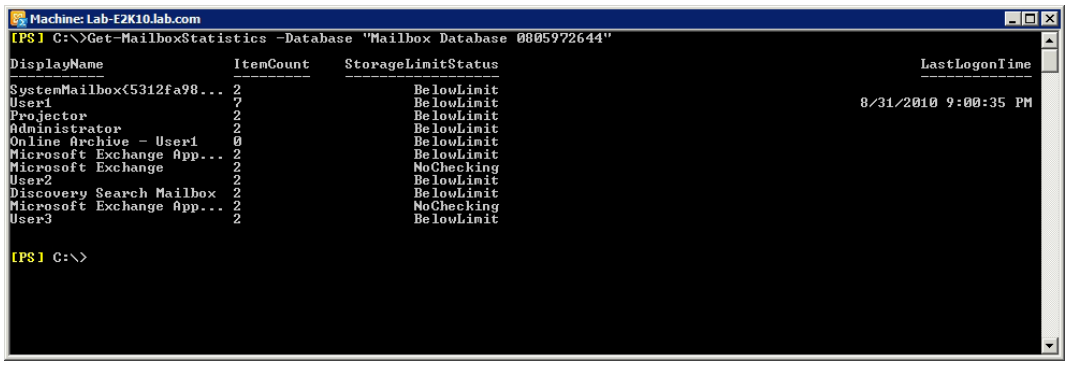

Figure 15: Get-MailboxStatistics Output

The only required attribute is the mailbox database name.

Another way in which you can get some statistics for a mailbox database is by using the Get-LogonStatistics command. The syntax used for this command is:

```
Get-LogonStatistics –Database <mailbox database name> | Format-List
```

The only required attribute is the –Database switch and the name of the mailbox database for which you want to collect statistics. The Format-List cmdlet is not required, but helps to make the cmdlet's output more easily readable. You will learn more about the Format-List cmdlet later in this domain.

When you run the Get-LogonStatistics cmdlet, Exchange returns several different pieces of information about the specified mailbox database. This information includes:

- The number and type of open items. For instance, Exchange reports the number of open messages and the number of open attachments.
- The number and type of operations. For instance, you will be able to view the number of messaging operations, table operations, transfer operations, progress operations, and total operations.
- The number of successful Remote Procedure Calls (RPCs).
- The names of the mailboxes within the database.
- Miscellaneous information regarding things like latency, client version, client IP address, and logon times.

In addition to providing database statistics, the Get-LogonStatistics cmdlet can also provide information specific to an individual server or to an individual mailbox. The commands used in doing so are:

```
Get-LogonStatistics –Identity <mailbox name>
Get-LogonStatistics –Identity <server name>
```

Public Folder Statistics

Just as Exchange 2010 is able to provide you with mailbox statistics, it is also able to provide statistics on public folders and on public folder databases. To retrieve the statistics for a public folder, you must use the Get-PublicFolderStatistics cmdlet. The syntax used by this cmdlet is as follows:

```
Get-PublicFolderStatistics –Identity \foldername | Format-List
```

The only required attribute for this cmdlet is the folder's identity. You must provide the full path for the folder. For instance, if you wanted to access statistics for a folder named \Domain1\Forms, then the cmdlet would look like this:

```
Get-PublicFolderStatistics –Identity \Domain1\Forms | Format-List
```

Once again, the Format-List cmdlet is optional, but is used to make the command's output easier to read. You will learn more about this cmdlet in a moment.

The Get-PublicFolderStatistics cmdlet provides basic statistical information about a public folder. Sometimes, however, you may find that you require information that is more granular. In these types of situations, you would use the Get-PublicFolderItemStatistics cmdlet.

The Get-PublicFolderItemStatistics cmdlet uses exactly the same syntax as the Get-PublicFolderStatistics cmdlet. Here is an example of how this cmdlet can be used:

```
Get-PublicFolderItemStatistics –Identity "\Domain1\Forms" | Format-List
```

The Get-PublicFolderItemStatistics cmdlet returns the following pieces of information for the items in the specified folder:

- The type of time
- The item's subject line
- When the item was last modified
- When the item was last accessed
- When the item was originally created
- Any attachments for the item
- The total message size

Format List / Format Table

In some of the previous sections, you have seen cmdlets ending in Format-List. The Format-List cmdlet is generally used in conjunction with another cmdlet that is expected to produce a lot of data. It tells Exchange to format the data as a list, in which each individual piece of data is displayed on a separate line. The Format-List cmdlet is sometimes abbreviated as FL.

An alternative to the Format-List cmdlet is the Format-Table cmdlet. Like the Format-List cmdlet, the Format-Table cmdlet is also used in conjunction with another command that is expected to produce a lot of data. The difference is that the output is formatted as a table.

Using a table is generally only practical for listing items and a limited number of attributes. For example, suppose that you were listing each user's contact information. A table may work best if you were only listing each user's name, department, and phone number. However, if you also wanted to include the user's address and the name of the user's supervisor, then you would probably want to use a list instead of a table because the requested information would be too long to fit on a single line.

The format table cmdlet is sometimes abbreviated as FT.

Perform Message Tracking

Once in a while a user will have a message go missing, and may ask an administrator what happened to the message. This is where message tracking comes into play. As the name implies, message tracking allows you to track down a message's whereabouts.

Message tracking is enabled by default in Exchange 2010, but you can disable it if your server is low on resources. Keep in mind that message tracking works differently depending on the server role, and therefore the method that you will use to disable message tracking depends on the server's role. To disable message tracking on a transport server, you would use this command:

```
Set-TransportServer <server name> -MessageTrackingLogEnabled:$false
```

To disable message tracking on a mailbox server, you would use this command:

```
Set-MailboxServer <server name> -MessageTrackingLogEnabled:$False
```

Notice that the two commands are identical, except that one uses the Set-TransportServer cmdlet while the other uses the Set-MailboxServer cmdlet. Be sure to pay attention to this distinction when taking the exam.

If you later decide to re-enable message tracking, you can use the same commands, but change $False to $True.

It is also worth noting that just as message tracking logging is enabled by default, so too is subject logging. Subject logging allows you to search for a message by querying the subject line. Subject logging can be very handy and is normally something that should remain enabled. Even so, some organizations like to disable subject logging as a way of reducing the volume of message logging data or of adhering to corporate privacy or security policies. Before I show you how to disable subject logging, please realize that subject logging only logs the message's subject line, not the message's body.

With that said, subject logging can only be enabled or disabled through the Exchange Management Shell. As with disabling message tracking, the command that you will use varies depending on the Exchange Server's role. To disable subject logging on a transport server, you would use this command:

```
Set-TransportServer <server name> -MessageTrackingLogSubjectLoggingEnabled $False
```

Likewise, if you wanted to disable subject logging on a mailbox server, you would use this command:

```
Set-MailboxServer <server name> -MessageTrackingLogSubjectLoggingEnabled $False
```

You can enable subject logging by using the two commands shown above, but changing $False to $True.

The Message Tracking Log Path

Although you must use the Exchange Management Shell to enable or disable the message tracking logs, you can use either the Exchange Management Shell or the Exchange Management Console to change the log's location.

If you want to use the Exchange Management Console to change the location of the message tracking logs, you can do so by navigating through the console tree to Server Configuration | Hub Transport (or Edge Transport), and then clicking the Properties link located beneath the name of the transport server. When the console displays the properties sheet, go to the Log Settings tab. Next, click the Browse button that is located next to the Message Tracking Log Path option. Select the new location in which you want to store the message tracking logs and then click Apply, followed by OK.

If you want to modify the message tracking log path from the Exchange Management Shell, then the cmdlet that you use will depend on whether you are modifying the path on a transport server or on a mailbox server. The commands that you would use are:

```
Set-TransportServer <server name> -MessageTrackingLogPath <local file path>

Set-MailboxServer <server name> -MessageTrackingLogPath <local file path>
```

The local file path must be enclosed in quotation marks. For example, the local log path might be:

```
"C:\Logs"
```

Message Tracking Log Sizes

As you can imagine, the message tracking logs can accumulate a lot of data – particularly in large organizations and on servers that process a high volume of mail. Thankfully, there are some ways that you can prevent the message tracking logs from growing out of control.

It is important to understand that Exchange does not lump all the message tracking data together. Instead, Exchange creates message tracking log files. Exchange continues to add data to a log file until the log has reached 10 MB in size (or reaches its maximum age), at which time a new log file is created.

Of course 10 MB is just the default log size. You can adjust the log file size by using the Set-MailboxServer or the Set-TransportServer commands (depending on the type of server that is being configured). In either case, the required parameters include the server name and the new message log size. Suppose, for instance, that you wanted to configure a mailbox server named Mailbox1 to support a log file size of 50 MB. To do so, you would use the following command:

```
Set-MailboxServer Mailbox1 –MessageTrackingLogMaxFileSize 50MB
```

To do the same thing on a transport server named Hub1, you would use this command:

```
Set-TransportServer Hub1 –MessageTrackingLogMaxFileSize 50MB
```

Keep in mind that these commands only control the maximum log file size. They have nothing to do with the total volume of logging data that is stored on the server. By default, Exchange will store 250 MB of logging data on an Exchange Server, although this also is configurable.

The message logging process works similarly to circular logging. When a message is sent or received, the tracking data is written to the current log file. Message tracking data continues to be written to the log file until it either reaches the maximum log file size or it reaches the maximum age. In either case, a new log file is started and used. Once the server accumulates 250 MB worth of logging data, then the oldest log file is deleted and a new log file is created.

You can change the maximum amount of logging data that can accumulate on a server by using the MessageTrackingLogMaxDirectorySize parameter in conjunction with the Set-TransportServer or Set-MailboxServer command. You would typically only increase the volume of logging data that is retained if the server processes so many messages that logging data is purged before it exceeds its useful lifespan.

Suppose that you wanted to increase the message tracking data retention from 250 MB to 500 MB. To do so, you would use one of the following commands (depending on the server role):

```
Set-MailboxServer <server name> -MessageTrackingLogMaxDirectorySize 500MB

Set-TransportServer <server name> -MessageTrackingLogMaxDirectorySize 500MB
```

Message Tracking Data Age

Throughout this section, I have mentioned that Exchange will purge message tracking data if it has exceeded its maximum age. By default, message tracking data is retained for 30 days. Transaction logs containing older data are automatically purged.

Keeping 30 days' worth of message tracking data should be sufficient for most organizations. However, you can increase or decrease the message age to meet your needs.

Changing the maximum age involves using either the Set-MailboxServer or the Set-TransportServer cmdlet, just as you have been using these cmdlets for other functions. In doing so, you would append the –MessageTrackingLogMaxAge switch and the new retention age. The one thing that you need to know about this command is that the maximum age needs to be entered in a specific format. This format looks like this:

```
Days.Hours:Minutes:Seconds
```

For example, suppose that you wanted to set the maximum age of the message tracking data to 60 days on a mailbox server named Mailbox1. To do so, you would use this command:

```
Set-MailboxServer Mailbox1 –MessageTrackingLogMaxAge 60.00:00:00
```

Manage Message Queues

When an Exchange Server receives a message, the message isn't immediately routed to its destination. Instead, the message is placed in a queue where it awaits processing. This allows messages to be processed in the order received.

The message queues are based on Extensible Storage Engine (ESE) databases. These are the same types of databases that are used for mailboxes and public folders. Unlike mailbox databases and public folder databases, however, you do not have to back up the message queues because messages typically pass through the queues so quickly that queue backups would be impractical.

Exchange Server 2010 makes use of five different types of queues. These queues include:

- **Mailbox Delivery Queue** – The mailbox Delivery Queue is unique to hub transport servers. A mailbox delivery queue acts as a repository for messages that are about to be sent to a mailbox server that resides in the same Active Directory site as the Hub Transport Server on which the queue resides. In larger organizations, a Hub Transport Server may contain several mailbox delivery queues, because a separate queue is created for each mailbox server in the Active Directory Site. Messages passing through a Mailbox Delivery Queue are sent to the mailbox server using Exchange encrypted RPC.
- **Poison Message Queue** – Sometimes when an Exchange Server failure occurs, one or more messages may become corrupted. If Exchange detects that a message may have become corrupted to the point that it could cause a queue to stall or cause other harm to the Exchange organization, the message is placed into the Poison Message Queue. A Poison Message Queue exists on the Edge Transport Server and on each Hub Transport Server. Typically these queues are hidden unless poison messages are present. If a message does make it into the Poison Message Queue, that message is held indefinitely until an administrator either deletes or releases the message.
- **Remote Delivery Queue** – Remote Delivery Queues are used for messages that are being sent to a remote server using SMTP. Remote Delivery Queues exist on both Hub Transport Servers and Edge Transport Servers. In either case, there may be several remote delivery queues. This is because Exchange creates a separate remote delivery queue for each destination. In the case of a Hub Transport Server, a separate queue is created for each remote Active Directory site. On Edge Transport Servers, a separate queue is created for each remote SMTP domain (and for each smart host). In order to prevent an accumulation of remote delivery queues, queues are dynamically created and deleted on an as needed basis.
- **Submission Queue** – Each transport server contains one submission queue. All messages enter the transport server through a receive connector and are placed into a submission queue. Once in the submission queue, Exchange looks at the message's recipient information, determines how the message needs to be routed, and places the message into the appropriate queue. In short, the submission queue's main job is to categorize messages.
- **Unreachable Queue** – Each transport server contains an unreachable queue. Any message containing recipients to whom the message cannot be routed is placed into the Unreachable Queue. When messages begin showing up in the Unreachable Queue, it is almost always the result of a configuration error.

You can view these queues through the Queue Viewer, which is accessible through the Exchange Toolbox. Notice in **Figure 16**, however, that the Queue Viewer only displays message queues that are currently active. For example, if no messages are in the Unreachable Queue, then that queue is not displayed.

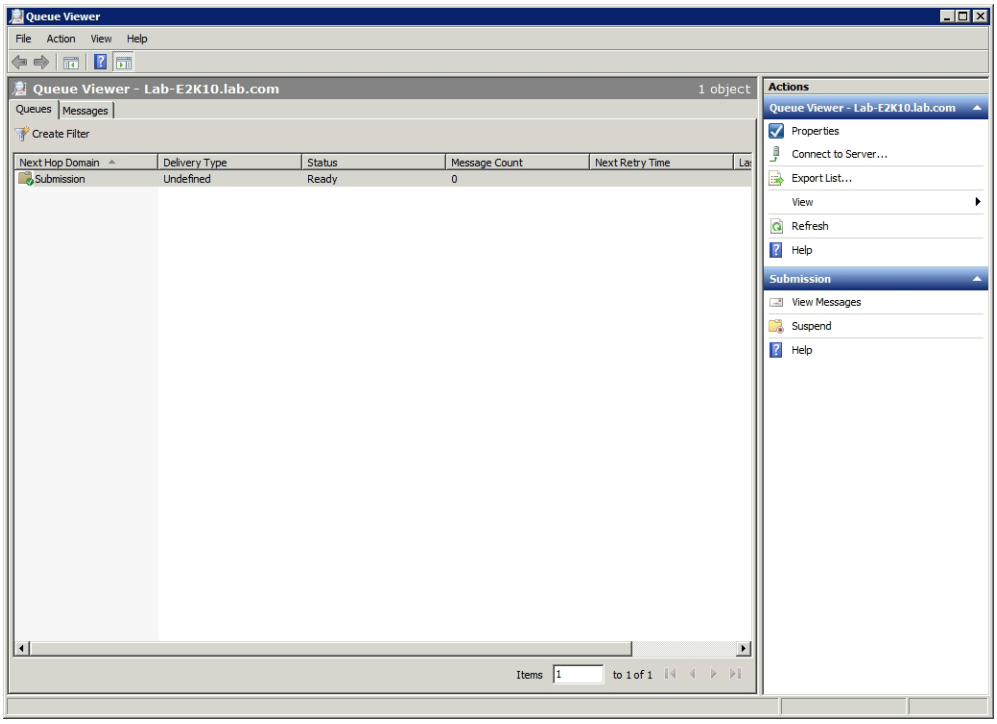

Figure 16: The Queue Viewer

You can monitor message queues through the Queue Viewer.

Resubmitting Queued Messages

When Exchange fails to route messages to their intended recipients, you can sometimes place those failed messages back into the submission queue, where Exchange will re-evaluate the messages and attempt to send them once again. You can resubmit messages from the following queues:

- Messages from the Unreachable Queue (so long as the messages are not in the suspended state)
- Messages in the Poison Mail Queue
- Messages in the Mailbox Delivery Queue that have a status of Retry
- Messages in the Remote Delivery Queue that have a status of Retry

If you want to resubmit messages that are located in one of the queues listed above, you can do so by using the Retry-Queue cmdlet. The exact syntax of the cmdlet will vary depending on what you are trying to accomplish. In most cases, you will need to know the name of the server and the name of the queue containing the messages that you want to resubmit.

At its simplest, you can enter the Retry-Queue cmdlet and provide the identity of the server and the queue from which you want to resubmit the messages. The command must be followed with an instruction to move forward with the resubmission. For example, suppose that you wanted to resubmit messages residing in the Unreachable Queue on a server named Server1. In this situation, you could submit the messages by entering the following command:

```
Retry-Queue –Identity "Server1\Unreachable" –Resubmit $true
```

In the command above, the –Resubmit $true parameter is necessary to make the resubmission begin.

The only problem with the command shown above is that it causes all of the messages in the queue to be resubmitted. This is fine if you have just corrected an issue which caused messages to accumulate in the Unreachable Queue and wanted to process those messages. However, you may occasionally want to only process certain messages from within a queue.

One common example of this is that you may want to resubmit all of the messages that have a status of Retry. You can accomplish this by adding a filter to the command used above. Of course, there will often be many different mailbox delivery queues and remote delivery queues, and it would take forever to manually resubmit the messages from each individual queue. A more efficient technique is to simply specify the name of the transport server containing the messages, rather than providing Exchange with the names of each individual queue.

For example, suppose that you had a server with many different mailbox delivery queues and remote delivery queues, and you wanted to resubmit any message with a Retry status, regardless of which queue it was in. To do so, you could use the following command:

```
Retry-Queue –Filter (Status –eq "Retry") –Server "Server1" –Resubmit $True
```

As you can see, the command shown above sets up a filter so that only messages with a status equal to (-eq) Retry are resubmitted. Rather than specifying a queue name, the command above simply provides the name of the server (Server1 in this case).

Backpressure Thresholds

If you have been working in IT for a while, you have no doubt seen servers which cease to function properly because they are low on hardware resources. Exchange 2010 contains a mechanism called backpressure monitoring that is designed to guard against resource deprivation.

Suppose, for example, an Edge Transport Server is low on system resources. Rather than rejecting connections to the server as Exchange 2007 would have, Exchange 2010 allows the connection, but throttles the inbound messages in an effort to prevent the server from running out of resources. In more extreme situations, Exchange may allow connections to the Edge Transport Server, but reject messages that are flowing across those connections.

Backpressure monitoring keeps tabs on five separate system resources:

- The amount of free disk space on the volume containing the message queue database
- The amount of free disk space on the volume containing the message queue transaction logs
- The number of uncommitted message queue database transactions
- The amount of memory consumed by the EdgeTransport.exe process
- The total amount of memory consumed by all system processes

Resource consumption is rated as high, medium, or normal for each of the resources listed above, and is based on various formulas. Whenever a resource is overused, Exchange takes action automatically. The action that is taken depends on which resource is being overused, and on whether the resource is slightly overused (medium) or significantly overused (high).

Although you can reconfigure some of the backpressure threshold values, Microsoft strongly advises against doing so, Therefore, the exam may require you to know what backpressure monitoring is, but you will not be expected to know how to reconfigure backpressure threshold values.

Monitoring ActiveSync

Depending on how many users have mobile devices synchronized to their Exchange mailboxes, and on factors such as the volume of mail each user sends and receives and attachment sizes, ActiveSync can place a significant load on a Client Access Server. As such, it is a good idea to monitor ActiveSync usage so that you can get a better idea of what the end user experience is really like.

ActiveSync is based on the Internet Information Services (IIS), which means that the ActiveSync logs are actually nothing more than IIS logs that are associated with the virtual directory used for ActiveSync.

Although the ActiveSync logs are not technically an Exchange Server feature, Exchange Server 2010 does provide a mechanism for creating ActiveSync reports that are based on the logging data. To create such a report, you must know the date range for which you want to export ActiveSync logging information.

You can create an ActiveSync report by using the Export-ActiveSyncLog cmdlet. This cmdlet requires you to provide a filename for the report that you are creating, as well as the start and end date for the report data. You must also provide an output path.

For example, suppose that you wanted to create a report that consisted of ActiveSync data from August 1, 2010. To do so, you would use the following command:

```
Export-ActiveSyncLog -Filename: "c:\Windows\System32\LogFiles\W2SVC1\ex080110.log"
-StartDate:"08/01/10" -EndDate:"08/01/10" -UseGMT:$false -OutputPath:"c:\logs"
```

The command shown above also contains a parameter called UseGMT which, if enabled, records events in Greenwich Mean Time, rather than your local time.

Besides just the basic ActiveSync report, there are several other reports available. These reports include:

- **Exchange ActiveSync Usage Reports** – This report keeps track of the objects that are synchronized. The report lists the object types, the number of objects of each type, and the total number of bytes associated with each object type.
- **Hits Report** – This report allows you to see the number of synchronization requests that are occurring per hour.
- **HTTP Status Report** – This report lists any HTTP errors that are occurring on the Client Access Server.
- **Policy Compliance Report** – A Policy Compliance Report lists the number of fully compliant, partially compliant, and non-compliant devices that are in use.
- **User Agent List** – This report lists each unique user and the operating system that is running on the user's mobile device.

Protocol Logging

Any time that an SMTP message is sent or received, there is an entire conversation that occurs between the sender and the recipient. Normally this conversation happens completely behind the scenes. But if mail flow problems occur, then knowing the details of the conversation can make the troubleshooting process easier. This is where protocol logging comes into play.

Protocol logging (which is disabled by default) is enabled or disabled at the connector level, As such, each send connector and each receive connector has its own set of protocol logs (assuming that you enable protocol logging for the connector).

You can enable protocol logging by using either the Exchange Management Console or the Exchange Management Shell. If you want to use the Exchange Management Console, then navigate through the console tree to Server Configuration | Hub Transport (or Edge Transport). Next, select the server that you want to modify, then click on the Receive Connectors tab. Finally, select the receive connector that you want to modify and then click the Properties link.

When the connector's properties sheet opens, go to the General tab and then use the Protocol Logging Level drop down box to either enable or disable protocol logging. Using the Verbose setting enables protocol logging, while the None setting disables it. Click Apply, followed by OK to save your changes.

You can also enable or disable protocol logging from the Exchange Management Shell by using the Set-ReceiveConnector or the Set-SendConnector command. To do so, you must specify the name of the connector that you want to enable logging for and set the logging level. For example, suppose that you wanted to enable verbose logging for a send connector named Domain1. To do so, you would use the following command:

```
Set-SendConnector "Domain1" -ProtocolLoggingLevel Verbose
```

The technique that I have just shown you will only work for connectors between Hub Transport Servers or Edge Transport Servers and the outside world. Exchange uses a different type of connector internally. This connector is called an intra-organization connector. Intra-organization connectors are created by default and are used to relay messages within the Exchange organization. Specifically, these connectors are used to relay messages:

- To other Hub Transport Servers
- To Exchange 2003 servers
- To Edge Transport Servers

Because the intra-organization connector is built in and created by default, you do not have to reference the connector by name. You are only required to specify the name of the Hub Transport Server to which the send connector is attached. To enable protocol logging for the intra-organization send connector, use the following command:

```
Set-TransportServer "<server name>" -IntraOrgProtocolLoggingLevel Verbose
```

As was the case with the message tracking logs, protocol logs are based on circular logging and you can customize the size and location of the log files.

By default, the Receive connector protocol logs are located at: C:\Program Files\Microsoft\Exchange Server\V14\TransportRoles\Logs\ProtocolLog\SMTPReceive.

The default location for the Receive connector protocol logs is C:\Program Files\Microsoft\Exchange Server\V14\TransportRoles\Logs\ProtocolLog\SMTPSend.

If you want to change the location of the log files, you can do so by using either the Exchange Management Console or the Exchange Management Shell. To use the Exchange Management Console, navigate through the console tree to Server Configuration | Hub Transport (or Edge Transport). Next, click the Properties link located beneath the server's name.

When the Properties sheet opens, go to the Log Settings tab. Go to the Protocol Log section, and then click on the Browse button next to either the Send Connector Protocol Log Path or the Receive Connector Protocol Log Path option. Specify the new log location path, and then click Apply, followed by OK. Keep in mind that you must keep the Send connector logs and the receive connector logs in separate locations.

If you want to use the Exchange Management Shell to change the protocol logging paths, you can do so by using the following commands:

```
Set-TransportServer <Server Name> -ReceiveProtocolLogPath "<new path>"

Set-TransportServer <Server Name> -SendProtocolLogPath "<new path>"
```

Adjusting the Protocol Log Size

As was the case with the message logs, the default size of the protocol logs are 10 MB. Even though logging is configured separately for each connector, you can run into situations in which multiple connectors share a common set of logs. Send and Receive connectors must be kept separate from each other, but all of the send connectors on a server (for which protocol logging is enabled) share a common set of logs. Likewise, all of a server's receive connectors (for which protocol logging is enabled) also share a common set of logs. As such, you may find that protocol logs tend to fill up quickly.

You can increase the size of a log file by using the Set-TransportServer command. In doing so, you must provide the name of the server on which you want to increase the log file size. You must also specify whether you are increasing the size of the send connector logs or the receive connector logs, as well as the new log size. For example, suppose that you had a server named Server1, and you wanted to increase the size of both the send connector and the receive connector logs to 50 MB. You could accomplish this by entering these commands:

```
Set-TransportServer Server1 -SendProtocolLogMaxFileSize 50MB

Set-TransportServer Server1 -ReceiveProtocolLogMaxFileSize 50MB
```

You can see an example of how these commands work in **Figure 17**. Notice that no output is displayed after the commands have been performed successfully.

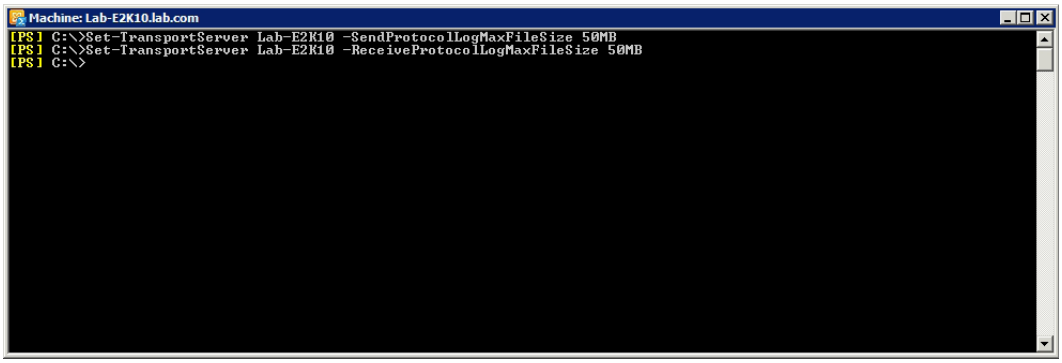

Figure 17: The Set-TransportServer cmdlet

The Set-TransportServer cmdlet does not produce any visible output.

The commands shown above control the maximum size of individual log files, but they have no impact on the overall volume of protocol logging data that is stored on the hub transport server. As you might expect, the maximum amount of protocol logging data stored on the server must be set separately for send connectors and for receive connectors. Therefore, if you decide to increase the size of the send connector folder to 500 MB and the size of the receive connector folder to 500 MB, then your protocol logging data could consume up to 1 GB of space on the server. I say that logging "could" consume up to 1 GB of space because it is possible for some log files to expire and be deleted before the server has a chance to accumulate a full gigabyte of data.

The commands used to set the maximum size of the protocol logs are based on the Set-TransportServer cmdlet. Suppose, for instance, that you wanted to set the maximum sizes of the send connector log folder and the receive connector log folder to 500 MB each. To do so, you would use the following commands:

```
Set-TransportServer <Server Name> -SendProtocolLogMaxDirectorySize 500MB

Set-TransportServer <Server Name> -ReceiveProtocolLogMaxDirectorySize 500MB
```

As was the case with the messaging logs, you can set the maximum age of the protocol logs. Once again, however, the maximum log age must be entered in the following format:

```
Days.Hours:Minutes:Seconds
```

Therefore, to set the maximum age of the send connector logs and the receive connector logs to 60 days on a server named Server1, you would use the following commands:

```
Set-TransportServer Server1 -SendProtocolLogMaxAge 60.00:00:00

Set-TransportServer Server1 -ReceiveProtocolLogMaxAge 60.00:00:00
```

Agent Logs

In Exchange Server 2010, both Hub Transport Servers and Edge Transport Servers have the ability to perform message hygiene by filtering inbound mail for spam. The spam filtering process is performed by several different agents:

- The Connection Filter Agent
- The Content Filter Agent
- The Edge Rules Agent
- The Recipient Filter Agent
- The Sender Filter Agent
- The Sender ID Agent

Each of these agents is capable of writing logging information to the agent logs. The agent logs are a compilation of information related to the actions of the agents listed above. Keep in mind, however, that the various agents may or may not actually write data to the agent logs, depending on the logging level that is in use. Logging levels are discussed a bit later in this domain.

Unlike message logs and protocol logs, you can't configure agent logs beyond simply enabling or disabling them and setting the logging levels (which will be discussed later).

Enabling agent logging also works differently from what you have seen with message logging and protocol logging. Rather than using an Exchange Management Shell cmdlet to enable logging, you are required to modify a configuration file. To do so, open Notepad and then use it to open C:\Program Files\Microsoft\Exchange Server\V14\Bin\EdgeTransport.exe.config.

After opening the configuration file, search for the <appSettings> section, as shown in **Figure 18**, and then modify the following line of code:

```
<add key=”AgentLogEnabled” value=”FALSE” />
```

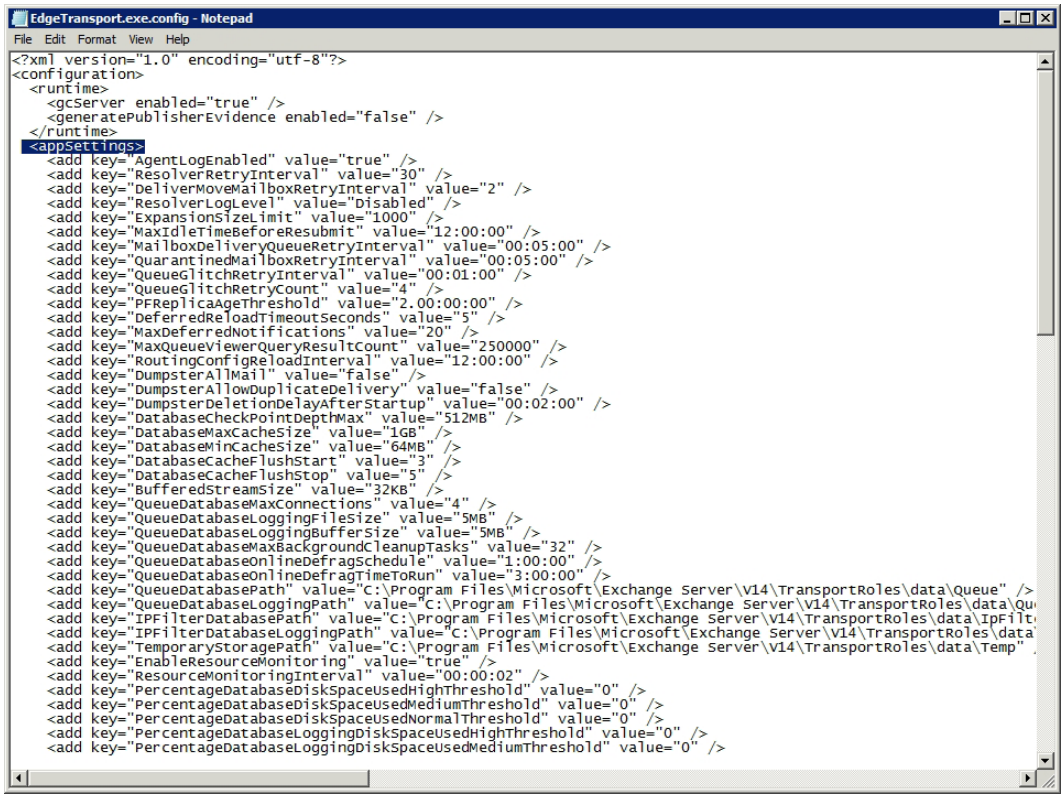

Figure 18: Writing AppSettings for EdgeTransport

The AppSettings section should be near the top of the file.

Setting the value to False disables agent logging, while a value of True causes agent logging to be enabled.

As different as agent logging is from message logging and protocol logging, it does use circular logging and many of the same default values as message logging and protocol logging. For example, the individual log files are 10 MB in size, and the maximum size of the agent logging folder is 250 MB. The agent logs also have a maximum age of 30 days.

You can find the agent logs at: C:\Program Files\Microsoft\Exchange Server\V14\TransportRoles\Logs\AgentLogs

Protocol Logging for POP3 and IMAP4

One of the best ways to troubleshoot issues with the POP3 and IMAP4 protocols is to enable protocol logging for POP3 and IMAP4. Protocol logging is not enabled for these protocols by default, and must be enabled or disabled through the Exchange Management Shell.

You can enable protocol logging for the POP3 protocol by entering the following command:

```
Set-POPSettings –ProtocolLogEnable –True
```

Likewise, you can enable protocol logging for the IMAP 4 protocol by entering this command:

```
Set-IMAPSettings –ProtocolLogEnable –True
```

If you want to disable protocol logging for either of these protocols, you can do so by using the commands shown above, but changing True to False.

Even though you can enable protocol logging for POP3 and IMAP4 through the use of Exchange Management Shell commands, enabling and disabling logging is the extent of what you can do from the command line. If you want to modify any of the protocol logging settings, you will have to do so by making modifications to the configuration files used by the protocols. The values that you can modify include:

- **AgeQuotaInHours** – This setting controls how frequently the logs will be recreated. By default the logs are refreshed every 24 hours.
- **SizeQuota** – The logs will automatically be recreated once they reach the specified size, unless they reach the age quota first. The default Size Quota is 10000000.
- **PerFileSizeQuota** – Exchange will create a new protocol log every time the current log file exceeds this threshold. The default value is 1000000.

In order to customize any of the values shown above, you must open the configuration file and then enable protocol logging from within the file. There is a separate configuration file for POP3 and for IMAP4. The configuration files, which can be opened in Notepad, are located at:

```
POP3 – C:\Program Files\Microsoft\Exchange Server\ClientAccess\PopImap\
Microsoft.Exchange.Pop3.exe.config

IMAP4 – C:\Program Files\Microsoft\Exchange Server\ClientAccess\PopImap\
Microsoft.Exchange.Imap4.exe.config
```

In either case, you can enable protocol logging by searching the configuration file for the <appSettings> section and then looking for the following line of code:

```
<addkey=”ProtocolLog” value=”false” />
```

A value of False disables protocol logging, while a value of True enables logging.

Configure Logging Levels

When Exchange Server 2010 experiences problems, the best way to diagnose those problems is often to examine the various logs. It is important to remember, however, that Exchange is capable of logging much more diagnostic information than is logged by default. You can control the amount of data that is logged by changing the diagnostic logging level.

You can adjust the logging level by using either the Exchange Management Shell or the Exchange Management Console. If you want to use the Exchange Management Console to adjust the logging level, you can do so by navigating through the console tree to Server Configuration | Mailbox, and then clicking the Manage Diagnostic Logging Properties link located in the Actions pane.

At this point, you must click on the individual service for which you want to adjust the logging level, and then click on the Configure link. If you decide that you need to reset Exchange to use only the default logging levels, you can do so by choosing the Reset All Services to Default Logging Levels option, and clicking Configure. In either case, you can complete the process by clicking Finish.

You also have the option of setting the logging levels through the Exchange Management Shell. The first thing that you should do is to use the Exchange Management Shell to retrieve a list of all of the various processes and their logging levels. You can do so by entering the following command:

```
Get-EventLogLevel
```

You can use the resulting list to retrieve the names of the various processes. This is important because you will have to include the process name when you change its logging level. You can change the logging level for a process by using the following command:

```
Set-EventLogLevel –Identity "<process name"> -Level <logging level>
```

As you can see, the command above requires you to specify a logging level. The logging levels that you can specify include:

Logging Level	Explanation
Lowest	This is the default logging level for each process. It only includes errors and events that are considered to be critical. Critical events have a logging level of 0.
Low	Any events with a logging level of 0 or 1 are logged.
Medium	Exchange logs all events with a level 3 or lower.
High	Exchange logs all events with a logging level of 5 or lower.
Expert	Exchange logs all events with a logging level of 7 or lower.

The Microsoft Exchange Best Practices Analyzer

Exchange Server is one of the most complex products that Microsoft manufactures. As such, Microsoft occasionally revises their best practices for Exchange as the Exchange Server team gains further experience with Exchange in real world settings.

Many years ago, Microsoft created a free tool called the Microsoft Exchange Best Practices Analyzer. This tool's job is to analyze the way that Exchange Server is configured, and then compare the configuration information to Microsoft's recommended best practices for Exchange. By doing so, Exchange administrators gain information that can help them to make their Exchange Server organizations more stable, more secure, and better performing.

Although you can download the Microsoft Exchange Best Practices Analyzer from Microsoft, it is already included with Exchange 2007 and Exchange 2010. You can find a link to the Microsoft Exchange Best Practices Analyzer in the Toolbox portion of the Exchange Management Console.

There are several different types of scans that you can perform by using the Best Practices analyzer. Those scans include:

- **Health Check** – A basic assessment of your organization's health.
- **Permissions Check** – A test that examines the permissions that are in place.
- **Connectivity Test** – A scan that is designed to spot network connectivity issues.
- **Baseline** – A comprehensive performance baseline that takes two hours to complete.

You can see the results from a sample health check in **Figure 19**. This health check was run against a lab server that had not been fully configured, so there were numerous issues reported. Still, this allows you to get a feel for the types of issues that the Best Practices Analyzer checks for.

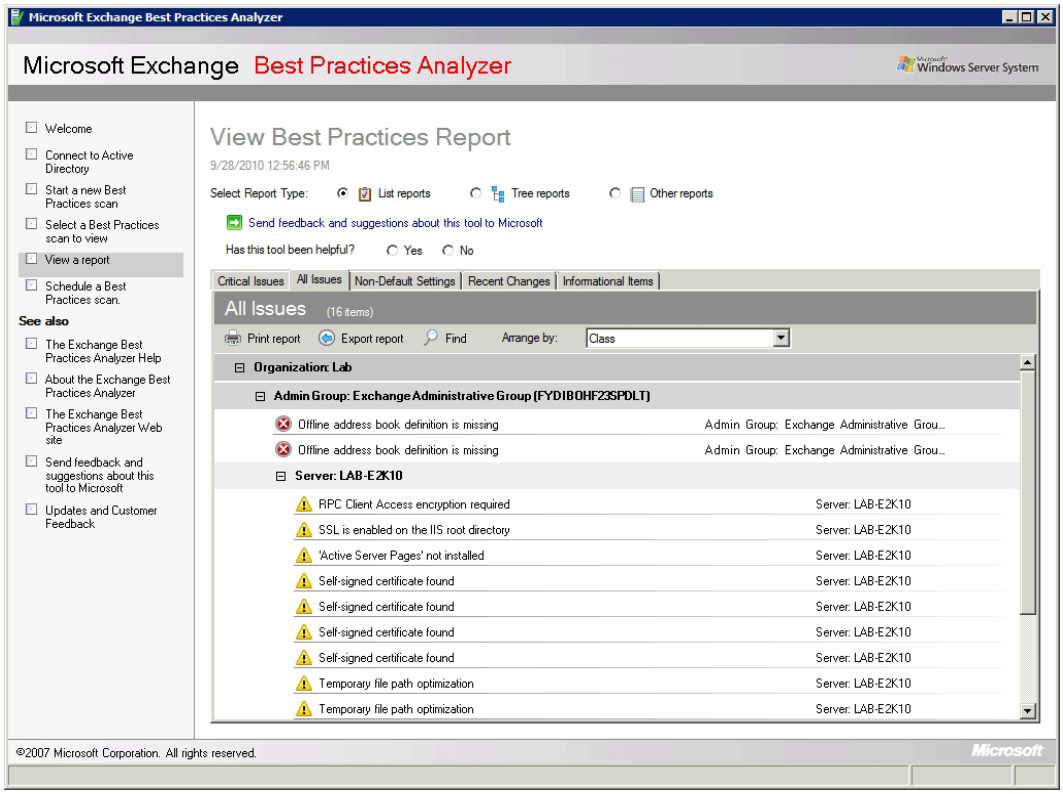

Figure 19: The Microsoft Exchange Best Practices Analyzer

This is what the results of a health check look like.

There are three main things that you should keep in mind regarding the use of the Best Practices Analyzer. First, whenever you run the Best Practices Analyzer, it gives you the chance to check for updates. You should allow the update check so that you can make sure that the tool is providing you with Microsoft's most recent recommendations.

The second thing that you need to remember about the Best Practices Analyzer is that because Microsoft changes their recommended best practices from time to time, you need to run the Best Practices Analyzer on a periodic basis. Many organizations make a point of running the Best Practices Analyzer on a monthly basis.

The third thing that you need to remember about the Best Practices Analyzer is that it is version specific. In other words, if you attempt to run the Exchange 2007 version of the Best Practices Analyzer against an Exchange 2010 Server, you will receive an error message because Exchange 2007 does not know how to handle Exchange 2010. You can, however, run the Exchange 2010 version of the Best Practices Analyzer against an Exchange 2007 server.

Domain 6: Implementing High Availability and Recovery

Database Availability Groups

In Exchange Server 2010, Microsoft has done away with the continuous replication and replaced it with a new feature called Database Availability Groups. A database availability group is a collection of mailbox servers that use failover clustering and continuous replication to provide fault tolerance against server, database, or network failures.

A database availability group can contain up to 16 mailbox servers, which allows for up to 16 different copies of each database. Of course, not every server within a database availability group has to have a copy of every database. You are free to mix and match databases as your needs dictate. For instance, if you have a database availability group consisting of four mailbox servers, you might have one database that is replicated to all four servers, while another database only exists on two of the four servers.

Creating Database Availability Groups

When you initially create a database availability group, it is nothing more than an Active Directory object. When you add the first mailbox server to the newly created database availability group, a couple of things happen.

First, Exchange updates the Active Directory so that the name of the newly added server becomes an attribute of the database availability group.

The second thing that happens is that Exchange automatically creates a failover cluster and all infrastructure components that go along with it.

A Database Availability Group can be created using either the Exchange Management Console or the Exchange Management Shell. To create a database availability group through the Exchange Management Console, navigate through the console tree to Organization Configuration | Mailbox. Next, click on the New Database Availability Group link found in the Actions pane. This will cause Exchange to launch a wizard that will guide you through the creation process, as shown in **Figure 20**.

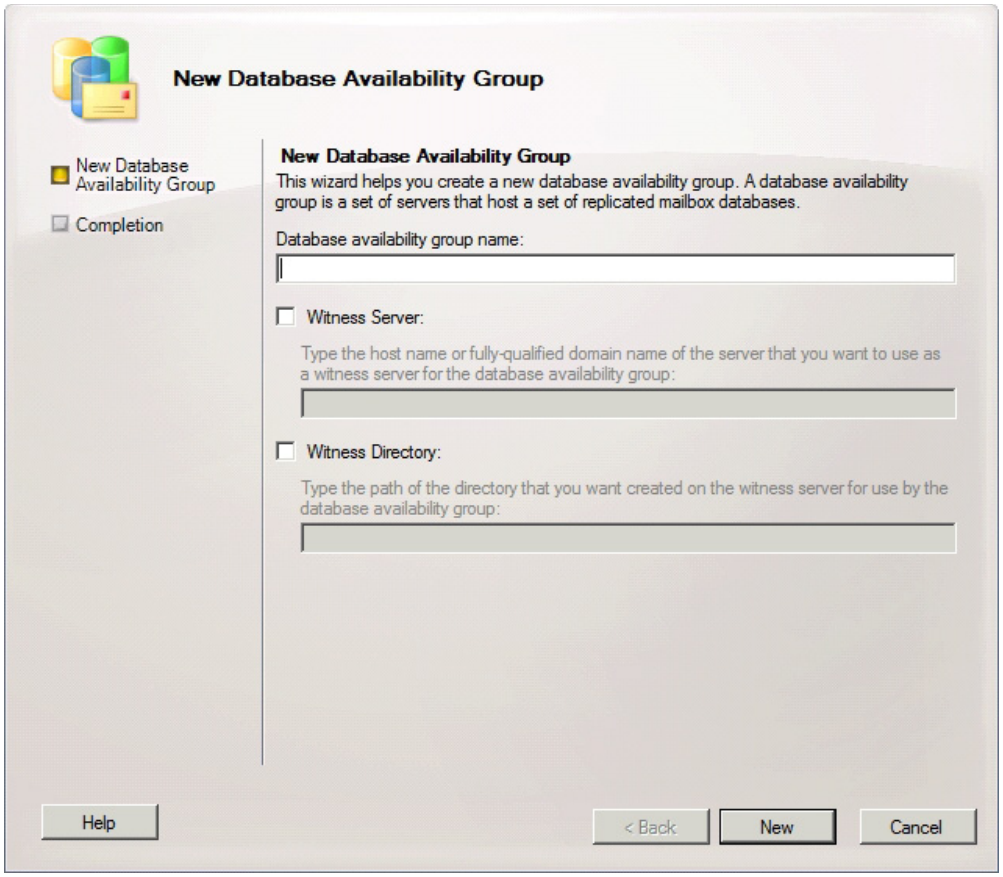

Figure 20: Creating a New Database Availability Group

You must provide a few key pieces of information when creating a Database Availability Group.

At this point, the wizard will prompt you for several pieces of information:

- **Database Availability Group Name** – The Database Availability Group Name works similarly to a computer name and is used solely for communications between database availability group members.
- **Witness Server** – As with Cluster Continuous Replication in Exchange 2007, Exchange 2010 database availability groups depend on a file share witness in order for the underlying failover cluster to maintain quorum.
- **Witness Directory** – This is the folder used to store file share witness related data.

Before I go on, there are a couple of things that you need to know about the witness server:

- You don't absolutely have to designate a witness server. If you leave the witness server option blank, then Exchange will try to designate a witness server automatically. Exchange looks for an Exchange 2010 hub transport server that does not have the mailbox server role installed.
- You can use a non Exchange 2010 server as a witness server. If you do, however, you must add the Exchange Trusted Subsystem universal group to the server's local Administrators group. Otherwise, Exchange will lack the necessary permissions to use the server as a file share witness.

- Once you have populated the wizard with the necessary information, click New followed by Finish to complete the process of creating the database availability group.
- If you choose to use the Exchange Management Shell to create a database availability group, you will have to provide the same basic information. The cmdlet used for creating a database availability group is New-DatabaseAvailabilityGroup. The cmdlet's syntax is:

```
New-DatabaseAvailabilityGroup –Name <database availability
group name> -WitnessServer <witness server name> -Witness
Directory <witness directory path>
```

An example of this command is as follows:

```
New-DatabaseAvvailabilityGroup –Name DAG –WitnessServer Hub1 –
WitnessDirectory C:\DAG
```

Configuring Database Availability Groups

After creating a database availability group, you can begin adding mailbox servers to the group. As you do, it is important to understand that database availability groups are dependent on failover clustering. As such, there are several things that you need to keep in mind before you begin adding mailbox servers to a database availability group:

- A mailbox server can only belong to one database availability group.
- Database availability group members must all run the same operating system.
- The only operating systems that support database availability groups are Windows Server 2008 (SP2 or higher) and Windows Server 2008 R2.

You can use either the Exchange Management Console or the Exchange Management Shell to add a mailbox server to a database availability group. To use the Exchange Management Console, navigate through the console tree to Organization Configuration | Mailbox. Next, choose the result pane's Database Availability Group tab. Right click on the database availability group to which you want to add a mailbox server, then choose the Manage Database Availability Group Membership option from the resulting shortcut menu.

At this point, Exchange will display a page that gives you the option of adding a mailbox server to or removing a mailbox server from the availability group. If you want to add a mailbox server, click on the Add button and then select the mailbox server that you want to add to the group.

If you want to remove a mailbox server from the database availability group, select the server that you want to remove and click the Delete icon. Keep in mind that before you can remove a server from a database availability group, you must remove all replicated database copies from the server.

In most cases, using the Exchange Management Shell to add or remove mailbox servers from a database availability group is really simple. You must use either the Add-DatabaseAvailabilityGroupServer or the Remove-DatabaseAvailabilityGroupServer cmdlet, then supply the name of the database availability group and the name of the mailbox server that you want to add or remove. Here are a couple of examples of how this cmdlet works:

```
Add-DatabaseAvailabilityGroupServer –Identity DAG –MailboxServer Server1

Remove-DatabaseAvailabilityGroupServer –Identity DAG –MailboxServer Server1
```

Occasionally, you may run into a situation in which a mailbox server within a database availability group needs to be taken offline for an extended period of time. In these types of situations, Microsoft recommends removing only the server's configuration. That way, the database availability group will be able to properly maintain quorum while the server is offline. To remove a mailbox server's configuration, you would use the same command that was shown above, but would append the –ConfigurationOnly switch. Here is an example of how the command works:

```
Add-DatabaseAvailabilityGroupServer –Identity DAG –MailboxServer
Server1 -ConfigurationOnly
```

Database Availability Group Networks

It is common for large database availability groups to span multiple Active Directory sites or multiple subnets. In these types of situations, you can split a database availability group into two or more networks so that the mailbox servers within the groups are accessed by clients in a logical manner.

You can create a database availability group network by using either the Exchange Management Console or the Exchange Management Shell. If you want to create a database availability group network using the Exchange Management Console, navigate through the console tree to Organization Configuration | Mailbox. Upon doing so, select the Database Availability Group tab within the results pane. Now, right click on the database availability group that you want to segment and then choose the New Database Availability Group Network command from the resulting shortcut menu. When you do, Exchange will launch the New Database Availability Group Network Wizard.

This one page wizard requires you to enter a few different pieces of information:

- **Network Name** – You must assign a name of 128 characters or less to the network that you are creating.
- **Network Description** – Although a description isn't required, providing a description for documentation purposes is a good idea.
- **Database Availability Group Network Subnets** – You must enter the group network subnets in IP Address / Bitmap format. For example, an IPv4 address might be entered as 192.168.1.0/24. IPv6 addresses are also acceptable. If you enter a network subnet that is already in use within a different database availability group network, then the subnet will be removed from that network.
- **Enable Replication** – This is a check box used to designate whether or not the specific subnet should be reserved for replication traffic. If the check box is selected, then MAPI traffic will not be allowed on the specified network.

After entering the required information, click New followed by Finish to create the network.

You can create a database availability group network through the Exchange Management Shell by using the New-DatabaseAvailabilityGroupNetwork cmdlet. The syntax is as follows:

```
New-DatabaseAvailabilityGroupNetwork –DatabaseAvailabilityGroup <database
availability group name> -Name <network name> -Description "<optional
description>" –subnets <subnets> -ReplicationEnabled:<$true or $false>
```

Here is an example of the command:

```
New-DatabaseAvailabilityGroupNetwork –DatabaseAvailabilityGroup
DAG –Name Network1 –Description "Sample Database Availability Group
Network" –Subnets 192.168.1.0/24 –ReplicationEnabled:$false
```

Add and Remove Database Copies

When you create a database availability group, one copy of each mailbox database is treated as the active copy. You can create up to 15 passive copies of a mailbox database (on 15 different mailbox servers) for a total of up to 16 database copies.

Before you can create a database copy, there are a few prerequisites that must be met:

- The original copy of the mailbox database will be treated as the active database copy by default. The active database copy must be mounted.
- The server that currently hosts the active database and the server that will host the database copy must be in the same database availability group.
- The database availability group must be in a healthy state.
- The mailbox database must not have circular logging enabled.

Incidentally, you can only create database copies for mailbox databases. Database availability groups do not support public folder databases.

You can use either the Exchange Management Console or the Exchange Management Shell to add a mailbox database console. If you want to use the Exchange Management Console, you must begin by navigating through the console tree to Organization Configuration | Mailbox. Next, select the Database management tab from the results pane. Right click on the database for which you want to create a copy, then choose the Add Mailbox Database Copy command from the resulting shortcut menu. When you do, Exchange will launch the Add Mailbox Database Copy Wizard.

The wizard only contains two fields that you have to populate:

- **Server Name** – You must specify the name of the mailbox server on which the database copy should reside.
- **Activation Preference Number** – This number reflects the order of preference when activating passive database copies. Databases with lower activation preference numbers take precedence when a passive copy of a database must be activated.

When you are done, click Add followed by Finish to create the database copy.

If you prefer to use the Exchange Management Shell to create a database copy, you can do so by entering the Add-MailboxDatabaseCopy cmdlet. The syntax is as follows:

```
Add-MailboxDatabaseCopy –Identity <database name> -Mailbox Server <mailbox
server name> -Activation Preference <activation preference number>
```

Here is an example of the command:

```
Add-MailboxDatabaseCopy –Identity DB1 –MailboxServer Domain12 –
ActivationPreference 2
```

Activating a Passive Database Copy

Occasionally, you may need to take the mailbox server that is hosting an active database copy down for maintenance. Before doing so, you will need to designate one of the passive database copies as the new active database copy. That way, the mailbox database can continue to function in the usual manner while the server is down.

You can activate a passive database copy by using either the Exchange Management Console or the Exchange Management Shell. If you want to use the Exchange Management Console, then begin the process by navigating through the console tree to Organization Configuration | Mailbox. Next, select the Database Management Tab, located in the results pane. Right click on the mailbox database that you want to work with, and then choose the Activate a Database Copy command from the resulting shortcut menu. When you do, Exchange will launch the Activate a Database Copy wizard.

At this point, you must click the Browse button, and then select the mailbox server containing the passive copy of the database that you want to activate. Choose whether or not you want to automatically mount the database, and then click the Move button to activate the database. When the process completes, click Finish to close the wizard.

Activating a passive database through the Exchange Management Shell is almost as easy as using the wizard. Before I show you how to do so, there are a couple of quick things that I want to point out.

- Each copy of a database has the same name. Therefore, database copies are usually referenced in database name \ mailbox server name format. When activating a database, however, the database name and the mailbox server name are entered separately.
- The Move-ActiveMailboxDatabase cmdlet uses a switch called MountDialOverride. This switch allows you to override the automatic mounting settings that have been previously assigned to the database. You will normally set this switch to a value of None.

With that said, you can move a database by using the Move-ActiveMailboxDatabase cmdlet. The syntax for this cmdlet is as follows:

```
Move-ActiveMailboxDatabase <mailbox database name> -ActivateOnServer
<mailbox server name> -MountDialOverride:<override settings>
```

Here is an example of how this command is used:

```
Move-ActiveMailboxDatabase DB –ActivateOnServer Domain13 –
MountDialOverride:none
```

Occasionally, you may discover that the database that you are trying to activate cannot be activated for various reasons. For example, if the database content index is not up to date, or if the index's state cannot be verified, then the database will not activate. To get around this problem, append the –SkipClientExperienceChecks switch to the Move-ActiveMailboxDatabase cmdlet.

Another common reason why a database may not mount is because there is an excessive lag. In other words, Exchange does not consider the database to be current because there are more than six transaction logs that have not yet been replayed. In such a situation, you could force the database to activate by appending the –SkipLagChecks switch to the Move-ActiveMailboxDatabase cmdlet.

Configuring Lag

Database copies are kept synchronized through the use of continuous replication. In other words, when a transaction log is filled up on the mailbox server that hosts the active copy of a database, a log shipping mechanism transfers a copy of the transaction log to the mailbox servers containing database copies. At that point, the transaction log is replayed against the database copy, bringing it up to date.

Because of the way that log shipping works, the active database and the database copies are never completely synchronized. Latency in the log shipping process ensures that database copies are always at least one transaction log behind the active database copy.

Sometimes it is desirable to build additional latency into a database copy. Suppose, for example, that your organization was hit with an e-mail virus. If the active database becomes infected, then all of the passive database copies will also become infected in a fairly short period of time.

To prevent situations like this, Exchange 2010 allows you to introduce a lag onto database copies. Doing so prevents transaction logs from being replayed on database copies until a certain length of time has passed.

When you create a database copy, there is no lag by default. You can, however, specify a lag time when you create the database copy. To do so, simply append the –ReplayLagTime switch to the Add-MailboxDatabaseCopy cmdlet, and then specify the desired lag time. For example, if you wanted to create a database copy with a three day lag time, then you could use a command like this:

```
Add-MailboxDatabaseCopy –Identity DB –MailboxServer Domain13 –
ReplayLagTime 3.00:00:00 –ActivationPreference 3
```

Now, suppose that you need to activate a lagged copy of a database. The first thing that you must do is suspend replication to the database. To do so, you can use the following command:

```
Suspend-MailboxDatabaseCopy DB\Domain13 –SuspendComment "Activating
lagged database copy"
```

Notice in the command above that we had to specify the database name as database name \ mailbox server name.

The next thing that you must do is to figure out which log files you want to replay and which ones you don't. You must exercise caution because a single database transaction can sometimes span multiple log files.

Once you have made your decision, back up the server and then delete the unwanted log files and the checkpoint file. Next, you must use ESEUTIL to perform a database recovery. The required command is:

```
ESEUTIL.exe /r eXX /a
```

In the command above, XX represents the log file prefix. In real life, log files are named E01, E02, E03, etc.

Once the command is complete, the database will be in a clean shut down state and can be used for recovery purposes.

High Availability for Non Mailbox Servers

Although Microsoft places a heavy emphasis on high availability for mailbox servers, it is also important to consider how you can make non mailbox servers highly available. After all, the failure of a client access server or a hub transport server could be just as disruptive as a mailbox server failure, unless other servers are in place to cope with the failure.

High Availability for Client Access Servers

In Exchange 2010, Client Access Servers handle all connections to Exchange mailbox servers. As such, it goes without saying that it is important to have redundant Client Access Servers.

The easiest way to accomplish this redundancy is to create a Client Access Server Array (CAS Array). A CAS array is a collection of Client Access Servers across which MAPI connections are distributed.

The first step in creating a CAS array is to set up your Client Access Servers in the usual manner. After doing so, you must load balance the servers. Exchange 2010 supports hardware load balancing as well as software load balancing through Windows' Network Load Balancing feature.

Next, you will have to designate an IP address that will be used to address the array as a whole. This IP address is known as the CAS Array's virtual IP address. You will also have to create a DNS entry that resolves the virtual IP address.

Now you are ready to actually create the CAS array. To do so, you will need to assign the array a friendly name as well as a fully qualified domain name that matches the DNS record that you created. The cmdlet used to create the CAS array is:

```
New-ClientAccessArray –Name "<friendly name>" –Fqdn "<fully qualified
domain name>" –Site "<site name>"
```

For example, the command might look something like this:

```
New-ClientAccessArray –Name "Domain1 CAS Array" –Fqdn "cas.Domain1.
com" –Site "Main Office"
```

If mailbox databases are already in place before the CAS array is created, then you will need to make each individual database aware of the new CAS array. To do so, you would use the following command:

```
Set-MailboxDatabase <database name> -RcpClientAccessServer "<cas array
fqdn>"
```

For example, the command might look like this:

```
Set-MailboxDatabase Domain1 –RcpClientAccessServer "cas.Domain1.com"
```

High Availability for Hub Transport Servers

Microsoft has designed Exchange 2010 Hub Transport Servers to automatically support load balancing and resilience. All you have to do is to deploy additional Hub Transport Servers within an Active Directory Site, and Exchange will do the rest.

There are four ways in which Hub Transport Servers provide resiliency:

- Hub Transport Servers are designed so that when messages are passed from one site to another, the messages are load balanced across the Hub Transport Servers in the destination site.
- Exchange 2010 mailbox servers automatically load balance outbound messages across any available Hub Transport Servers within the Active Directory site.
- Unified Messaging servers automatically load balance traffic across all of the available Hub Transport Servers within a site.
- Edge Transport Servers will automatically load balance inbound SMTP traffic across all of the Hub Transport Servers residing within the site in which the edge server has been subscribed.

High Availability for Edge Transport Servers

Exchange does not offer any high availability features for Edge Transport Servers. Microsoft's recommendation is that you deploy multiple Edge Transport Servers in parallel. After doing so, you can use the Network Load Balancing service to distribute the workload across the servers. Microsoft also recommends configuring your MX record with the IP address of each Edge Transport Server so as to achieve DNS round robin based load balancing.

Disaster Recovery for Exchange 2010

When an Exchange 2010 server fails, recovering the server is relatively easy. This is because almost all of the configuration information that Exchange uses is stored in the Active Directory. Therefore, if you have to rebuild a failed Exchange Server, all you have to do is to install the Exchange Server binaries and then associate the server with the configuration information found in the Active Directory.

This basic technique will work for all Exchange Server roles, except for the Edge Transport Server role. Having said that, there are some additional considerations for mailbox servers. Those considerations will be addressed later on. For right now, I want to show you the basic steps involved in recovering an Exchange Server.

Before you can begin the recovery process, there are several prerequisites that must be met:

- The server on which you are performing the recovery must be running the same operating system as was running on the failed server. Additionally, the new server should be running the same operating system service pack level as the failed server.
- The new server must have the same name as the failed server.
- If you are recovering a mailbox server, then the same drive letters must exist on the new server as existed on the failed server.

To perform the actual recovery process, begin by resetting the failed server's computer account in the Active Directory. It is critically important that you reset the account rather than deleting and recreating it. Otherwise, all of the Exchange Server configuration information will be lost.

Once you have reset the computer account, go ahead and install the operating system and the service packs onto the new server. Be sure to assign the new server exactly the same name as what the failed server was using.

At this point, you should join the new server to the domain that was used by the failed server. After doing so, you can install any prerequisite operating system components required by Exchange.

Finally, insert your Exchange 2010 installation media, open a Command Prompt window, and enter the following command:

```
Setup /m:recoverserver
```

Disaster Recovery for Mailbox Servers

You can recover a failed mailbox server using the technique that was discussed in the previous section. Keep in mind, however, that this technique will only restore the server's configuration. It will not recover the databases. For that, you will have to restore a backup (unless the database files still exist on the server).

In Exchange Server 2007, you were able to perform either a streaming backup or a Volume Shadow Copy Service (VSS) backup. A VSS backup is a point in time, snap shot backup. Exchange 2010 does not support streaming backups. VSS backups are the only supported backup technology.

Furthermore, there are some special considerations that need to be taken into account if you are backing up a database with multiple copies. For such databases you can only backup the active copy of the database using the built-in VSS writer. There are, however, commercial backup applications (such as Microsoft's System Center Data Protection Manager) that can backup a passive mailbox server.
If you do decide to backup a passive mailbox database, then you should know that you will not be able to restore it directly to the server that is hosting the passive database copy. The only way to perform a VSS restoration of such a backup is to restore the backup to an alternate location, suspend replication to the passive database copy, and then perform a file level copy of the recently restored database and its log files from its current location to the normal location.

The Recovery Database

Exchange 2007 supported single item recovery through the use of a recovery storage group. Because Exchange 2010 does not use storage groups, the Recovery Storage Group feature has also been removed. Instead, this feature has been replaced with a new feature called a recovery database.

A recovery database is useful in situations in which you need to restore an individual mailbox. In these types of situations, you would not want to restore the entire mailbox database, because doing so would cause live data to be overwritten. Instead, you can restore a backup to a recovery database, and then extract the specific mailbox data that you need without fear of interfering with production data.

There are several things that you must remember about recovery databases:

- A recovery database can only be used to recover Exchange 2010 mailbox databases.
- The target mailbox to which data from the recovery database will be extracted must be in the same Active Directory forest as the server containing the recovery database.
- You must create a recovery database prior to beginning the restore operation.
- The volume containing the recovery database must have adequate free space to accommodate the mailbox database that you are restoring and its log files.
- After restoring a database, you must use ESEUTIL /R to put the database into a clean shutdown state.

You can create a recovery database by using the New-MailboxDatabase cmdlet. The cmdlet works just as it would if you were creating any other type of database, but you must specify the –Recovery switch. For instance, if you wanted to create a recovery database named RecoveryDB on a mailbox server named Server1, you could do so by using the following command:

```
New-MailboxDatabase -Recovery -Name RecoveryDB -Server Server1
```

You can see an example of how the process works in **Figure 21.**

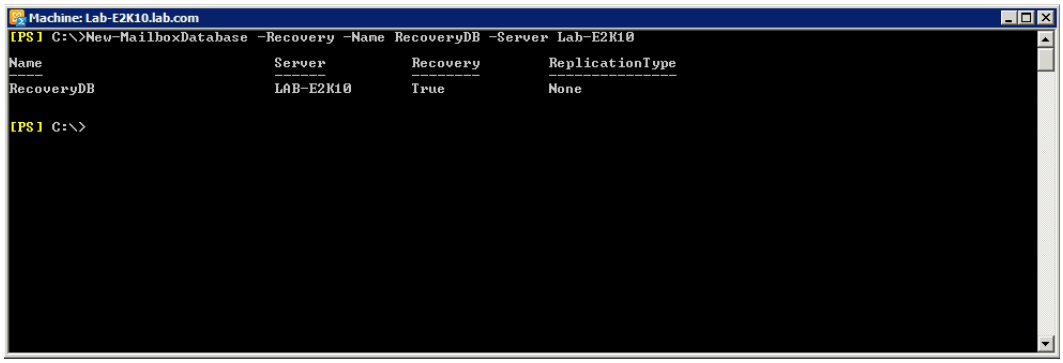

Figure 21: The New-MailboxDatabase cmdlet

You must use the New-MailboxDatabase cmdlet to create a recovery database.

As with any other database, you can use the –EdbFilePath and the –LogFolderPath switches to specify the paths that will be used by the database and its transaction logs.

After you have restored a backup to a recovery database, put the database into a clean shutdown state and mounted the database, you can restore mailboxes or mailbox data by using the Restore-Mailbox cmdlet.

There are quite a few variations of this cmdlet. At its simplest, you can specify the name of the mailbox that you want to restore and the name of the recovery database that you want to use. For example, if you wanted to restore the mailbox for a user named JohnDoe from a recovery database named RecoveryDB, you could do so with this command:

```
Restore-Mailbox -Identity JohnDoe -RecoveryDatabase RecoveryDB
```

You can use a similar command to perform a bulk restoration of all of the mailboxes found in the recovery database. To do so, you must specify the name of the production database that should be cross referenced against the recovery database. For example, suppose that you wanted to restore the mailboxes for every user in a database named Domain1DB. You could do so by using this command:

```
Get-Mailbox -Database Domain1DB | Restore-Mailbox -RecoveryDatabase RecoveryDB
```

Sometimes you may be required to restore individual mailbox items rather than an entire mailbox. In these types of situations, you can still use the Restore-Mailbox cmdlet. The difference is that you must add some filtering criteria as a way of controlling what actually gets restored. As a best practice, you should also create a folder within the mailbox and place the restored data into the folder.

To see how this works, suppose that you wanted to restore all of the messages in JohnDoe's mailbox that contain the word Domain1, and you wanted to place those messages in a mailbox folder called Recovery. You could accomplish this by using the following command:

```
Restore-Mailbox -Identity JohnDoe -RecoveryDatabase RecoveryDB -
ContentKeywords "Domain1" -TargetFolder Recovery
```

Dial Tone Recovery

Exchange Server 2010 databases can grow to become quite large. During a disaster recovery situation, you may find that it takes many hours to fully restore all of the mailbox databases. In these types of situations, you may find it useful to take advantage of an Exchange 2010 feature called Dial Tone Portability.

Dial Tone Portability gives you the option of creating temporary mailboxes for the users whose mailboxes you are restoring. That way, the users are able to continue to send and receive mail while you are working on restoring the user's data.

When you use this approach, the database containing the empty mailboxes is called the dial tone database, and the recovery process is known as a dial tone recovery. When you perform a dial tone recovery, the Autodiscover service automatically redirects Outlook clients to the dial tone database.

Dial tone recovery builds on the recovery database techniques covered in the previous section. Microsoft's preferred method for performing a dial tone restore involves creating the dial tone database directly on the server to which you are restoring the failed database. Having the dial tone database located directly on this server makes the recovery process more efficient once the database restoration completes.

Sometimes, a server may fail catastrophically and need to be rebuilt. In these types of situations, it is possible to create the dial tone database on an alternate server. After doing so, you can rebuild the failed server and then use database portability to move the dial tone database to the failed server once it is functional.
A third option is to create a dial tone database on an alternate server, and then restore the backup to the same server that is housing the dial tone database. This method is used when the original server experiences a hardware failure and cannot be repaired.

The first step in performing a dial tone recovery is to create a dial tone database on your server of choice. Suppose, for instance, that you wanted to call your dial tone database DTDB. To do so, you would use the following command to create the database:

```
New-MailboxDatabase –Name DTDB –EDBFILE D:\DialTone\DTDB.EDB
```

Once you have created the dial tone database, you must rehome the user mailboxes. This is the process that creates empty mailboxes within the dial tone database. The command used in this process requires you to specify the name of the database that was previously being used, as well as the name of the dial tone database. For instance, if the user mailboxes were previously stored in a database named Domain1, and the dial tone database is named DTDB, then you would use the following command:

```
Get-Mailbox –Database Domain1 | Set-Mailbox –Database DTDB
```

Once you have rehomed the user mailboxes, you must mount the dial tone database. You can do so by using this command:

```
Mount-Database –Identity DTDB
```

At this point, you must create a recovery database and restore your backup to it. After doing so, you should copy any existing log files from the failed database to the database folder that is used by the recovery database.

The next step in the process is to mount and then immediately dismount the recovery database. You can do so by entering the following commands:

Mount-Database –Identity RecoveryDB
Dismount-Database –Identity RecoveryDB

Now, move the recovery database and its log files to a safe location. When you are done, dismount the dial tone database. You can do so by entering this command:

Dismount-Database DTDB

Now, move the dial tone database and its log files into the recovery database folder. You must also move the database and log files that you stored in a safe location into the dial tone database folder. You can now mount the dial tone database by using the following command:

Mount-Database –Identity DTDB

At this point, your users should have access to all of the data that was restored, and they should be able to send and receive messages. The last step in the recovery process is to move the messages from the recovery database (which was previously the dial tone database) into the production database. You can accomplish this by entering these commands:

Mount-Database –Identity RecoveryDB
Get-Mailbox –Database DTDB | Restore-Mailbox –RecoveryDatabase RecoveryDB

When you are done, you can clean up your work by dismounting and deleting the recovery database. This can be accomplished through the use of the following commands:

Dismount-Database –Identity RecoveryDB
Remove-MailboxDatabase –Identity RecoveryDB

Merging PST Data

Just as you can merge data from a recovery database into a user's live mailbox, you can also merge data from a PST file into a user's mailbox. Doing so involves the use of the Import-Mailbox cmdlet.

At its simplest, the Import-Mailbox cmdlet requires you to specify the name of the Exchange mailbox and the name and path of the PST file. For example, if you wanted to merge PST data for a user named JohnDoe, you could use a command similar to this one:

Import-Mailbox –Identity JohnDoe@Domain1.com –PSTFolderPath C:\files\JohnDoe.pst

The Import-Mailbox cmdlet also allows you to perform bulk imports of PST data. The catch is that each of the PST files that you are importing must have a name matching the user's alias. For example, if you wanted to import PST data for a user named JohnDoe, then the PST file would have to be named JohnDoe.pst.

After you have verified that all of the PST files that you want to import are named appropriately, you must place all of the PST files into a common folder. After doing so, you can perform a bulk import by using the following command:

```
Dir C:\PSTs | Import-Mailbox
```

You also have the option of doing some filtering when you import PST data. For example, suppose that you only wanted to import PST data for the users in an organizational unit named Finance, and you only wanted to import messages that were received after August 1, 2010. To do so, you could use this command:

```
Get-Mailbox –OrganizationalUnit Finance | Import-Mailbox –StartDate
08/01/2010 –PSTFolderPath C:\PSTs
```

Deleted Item Retention

When a user deletes a message, the user has a couple of options for recovering the message. The first option is to retrieve the message from the Deleted Items folder. However, if the message has already been purged from the Deleted Items folder, then the next option is to use the Deleted Item Retention feature.

When an item is purged from a user's Deleted Items folder, it is actually moved into a hidden area of the user's mailbox. The mailbox contains a hidden folder called Non-IPM. This folder contains a hidden sub folder called Recoverable Items. In turn, the Recoverable Items folder contains three hidden folders named Deletions, Versions, and Purges.

By default, Exchange 2010 retains deleted items in the Recoverable Items folder for 14 days. This threshold is adjustable on either a per mailbox or a per database level. To set the retention threshold on a mailbox database, you can use the following command:

```
Set-MailboxDatabase <database name> -DeletedItemRetention <number of days>
```

You can see an example of how this command works in **Figure 23**.

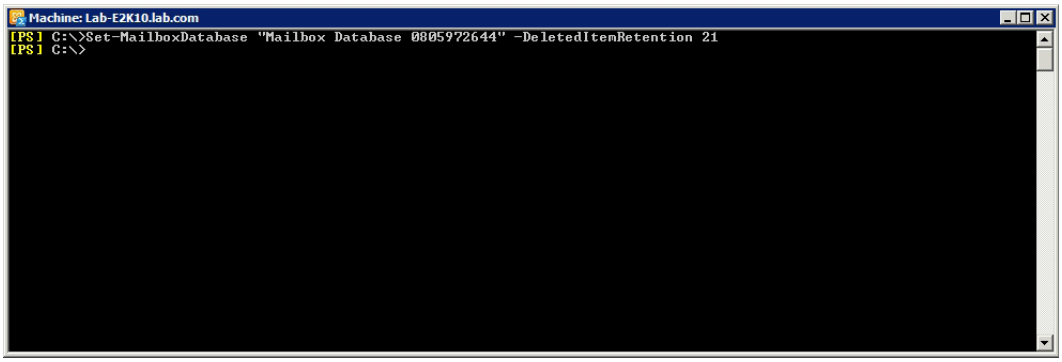

Figure 23: The Set-MailboxDatabase cmdlet

The Set-MailboxDatabase cmdlet is used to specify a deleted item retention period for a mailbox database.

To set the retention period for an individual mailbox, you would use the Set-Mailbox cmdlet as shown below:

```
Set-Mailbox <mailbox name> -RetainDeletedItemsFor <number of days>
```

Regardless of how the retention settings are configured, calendar items are retained for 120 days.

To recover a deleted item, someone who has been assigned the Discovery Management Role opens the Exchange Control Panel and verifies that the item that needs to be recovered still exists. After doing so, they must use the Export-Mailbox cmdlet to export the item from the discovery mailbox to the end user's mailbox.

Deleted Mailbox Retention

When you delete a mailbox in Exchange Server 2010, the mailbox is not permanently deleted. Instead, Exchange retains the mailbox until the specified retention period (30 days by default) expires. At that point, the mailbox is deleted.

Mailbox retention periods are configured on a per mailbox database basis. You can adjust the mailbox retention period for a mailbox database by using the Set-MailboxDatabase cmdlet. For example, suppose that you wanted to set the mailbox retention period to 60 days on a mailbox database named Domain1.

To do so, you would use this command:

```
Set-MailboxDatabase –Identity Domain1 –MailboxRetention 60
```

You can see an example of how this command works in **Figure 24**.

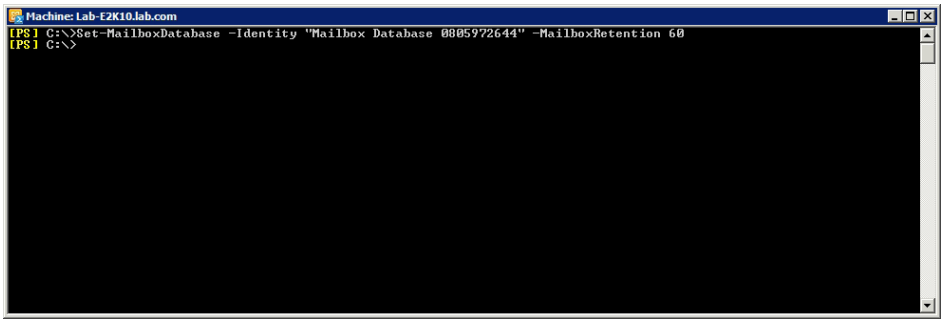

Figure 24: Using Set-MailboxDatabase to Specify a Retention Period

The Set-MailboxDatabase cmdlet is used to specify the retention period for deleted mailboxes.

The reason why Exchange retains deleted mailboxes is because deleted mailbox retention makes it possible to recover a deleted mailbox without having to restore a backup. To recover a deleted mailbox that is still within its retention period, you must treat the deleted mailbox as a disconnected mailbox.

Disconnected Mailboxes

As explained in the previous section, when you delete a mailbox, the mailbox isn't completely deleted until the mailbox retention period expires. Instead, the deleted mailbox becomes disconnected. A disconnected mailbox is Exchange speak for a mailbox that has become disassociated with a user account. A disconnected mailbox can be recovered by connecting it to a user account. Otherwise, the disconnected mailbox will be purged at the end of the retention period.

You can connect a disconnected mailbox by using either the Exchange Management Console or the Exchange Management Shell. To connect a disconnected mailbox through the Exchange Management Console, navigate through the console tree to Recipient Configuration | Disconnected Mailbox. Next, select the mailbox that you want to connect, and then click on the Connect link, located in the Actions pane. This will cause the console to launch a wizard that guides you through the connection process.

The wizard's initial screen asks you to select a mailbox type for the mailbox that you are connecting. The options available to you include:

- **User Mailbox** – A regular mailbox owned by a user.
- **Room Mailbox or Equipment Mailbox** – A mailbox used for scheduling resources.
- **Linked Mailbox** – A mailbox that is used by a user from an external Active Directory forest.

As you make your selection, keep in mind that you can only link a room mailbox or an equipment mailbox to a user account that has been disabled.

After you have made your selection, click Next and you will be taken to the wizard's Mailbox Settings page. This page requires you to specify the user account that should be linked to the mailbox. Your options include:

- **Matching User** – Choosing this option will cause Exchange to automatically look for a user account with a name that matches that of the mailbox.
- **Existing User** – Use this option to manually specify a user other than the matching user.

Additionally, there are a few other options that you can set:

- **Alias** – You can use this text box to enter the mailbox alias.
- **Managed Folder Mailbox Policy** – You have the option of selecting this check box and choosing a managed folder mailbox policy for the mailbox.
- **Exchange ActiveSync Mailbox Policy** – You have the option of selecting this checkbox and choosing an ActiveSync Mailbox Policy for the mailbox to use.

If you have chosen to connect to a linked mailbox, then the wizard will now display the Master Account page. This page requires you to provide several pieces of information:

- **Trusted Forest or Domain** – The name of the trusted forest or domain containing the master account.
- **Use the Following User Account to Access Linked Domain Controller** – You must provide the credentials for an account that has access to the linked domain controller.
- **Linked Domain Controller** – You must choose a domain controller containing the account that you want to use as a master account.
- **Linked Master Account** – This is the account that you want to link to the mailbox.

You should now be taken to the wizard's Connect Mailbox page. This page gives you a chance to review the settings that you have entered. Assuming that everything looks good, click the Connect button. Once the mailbox has been connected, click Finish to complete the process.

Using the Exchange Management Shell to connect a user to a disconnected mailbox requires you to provide the same information that was required by the Exchange Management Console. The process involves using the Connect-Mailbox cmdlet, but the command's exact syntax varies depending on the type of account that you are connecting.

At its simplest, this cmdlet requires you to provide the identity of the disconnected mailbox, the name of the database containing the mailbox, and the name of the user account that the mailbox should be connected to. For example, suppose that you wanted to reconnect a user named JohnDoe to a mailbox named JohnDoe that is located within a mailbox database named Domain1. To do so, you could use the following command:

```
Connect-Mailbox –Identity "JohnDoe" –Database Domain1 –User "JohnDoe"
```

The process works similarly if you are connecting an equipment mailbox or a room mailbox. The only difference is that you must tell Exchange that the mailbox is an equipment or a room mailbox. Suppose, for instance, that you wanted to connect a user named Projector1 with an equipment mailbox located in the Domain1 database named Projector1. To do so, you could use this command:

```
Connect-Mailbox –Identity "Projector1" –Database "Domain1" –Equipment
–User "Projector1"
```

The process works exactly the same for a room mailbox, except that you would use the –Room switch instead of the –Equipment switch. For example, a command used for linking a room mailbox might look something like this:

```
Connect-Mailbox –Identity "Room1" –Database "Domain1" –Room –User "Room1"
```

Finally, you can use the Connect-Mailbox cmdlet to connect a linked mailbox. The process works similarly to what you have already seen, except that you have to specify the name of a linked domain controller and a linked master account. Here is an example of a command that might be used to connect a linked account:

```
Connect-Mailbox –Identity "JohnDoe" –Database "Domain1" –
LinkedDomainController DC1 –LinkedMasterAccount JohnDoe@
Domain2traders.com
```

Rebuilding an Edge Transport Server

The requirements for recovering an Edge Transport Server after a failure are unique, because unlike other Exchange Server roles, the Edge Transport Server does not store its configuration in the Active Directory database. Instead, the Edge Transport Server stores its configuration in the Active Directory Light Weight Directory Service (AD LDS), which resides locally on the server.

The most effective way of backing up an Edge Transport Server is to perform a full, system state backup. Many organizations prefer to simply export the edge configuration. The advantage of doing so is that it is much faster to export the edge configuration information than it is to perform a full system state backup. The disadvantage is that a full system state backup allows the entire server to be recovered in a single step, while an edge configuration file is something that must be imported after a server has been manually rebuilt.

Microsoft provides a script for exporting the edge configuration. You can find the script in the following folder:

```
C:\Program Files\Microsoft\Exchange Server\V14\Scripts
```

To export the edge configuration information, open the Exchange Management Shell, navigate to the folder listed above, and run the following command:

```
.\ExportEdgeConfig.ps1 –CloneConfigData C:\EdgeConfig.xml
```

The command shown above starts with ./ which is how you tell PowerShell that you want to run the script. After the ./ we are providing the name of the script, a switch indicating that we want to clone the configuration data, and a path and file name. You can use any path and file name that you like. In this case, we are creating a file named EdgeConfig.xml and placing it in the C:\ folder.

If you have to rebuild an edge transport server (without the aid of a full system state backup), you will have to install Windows and then install the Edge Transport Server role. From there, you would open the Exchange Management shell, navigate to the server's scripts folder, and run the following command:

```
.\ImportEdgeConfig.ps1 –CloneConfigData C:\EdgeConfig.xml
```

As you can see, this command is almost identical to the previous command except for the fact that we are importing the edge configuration rather than exporting it. Before this command will work, the edge configuration file must be copied to the server in the path that you specify within the command.

Domain 7: Configuring Message Compliance and Security

Domain 7 focuses on security and compliance, with a heavy emphasis on records management. The basic idea behind records management is that in any organization, there are many different types of messages that are sent and received. Some of these messages, such as those dealing with business strategy or financial transactions, may need to be retained for a specific length of time to ensure compliance with various regulations or simply to comply with the organization's own internal policies. Other messages, such as personal messages or newsletters, may not require any long term retention at all. As such, establishing a blanket policy that requires all mail to be retained increases the demands placed on the storage subsystem beyond what is really necessary.

Message Records Management

Exchange Server 2010 contains a set of features that are collectively known as Message Records Management, or MRM. Message Records Management is designed to help organizations to differentiate between various types of e-mail messages, and then retain the various message types for specific lengths of time according to company policy.

Managed Folders

One of the most important concepts for you to understand regarding the use of Message Records Management is that of managed folders. That is because retention policies can be applied on either a per mailbox or on a per folder basis.

A managed folder is really nothing more than just an Active Directory representation of the folders that make up a user's mailbox. Exchange defines two different types of managed folders:

- **Managed Default Folders** – Managed Default Folders are managed folders that are built into Exchange. This includes things like the Inbox, Deleted Items, and Sent Items folders.
- **Managed Custom Folders** – Managed Custom Folders are folders that you create for the end users. Managed Custom Folders are located beneath a folder called Managed Folders within the folder hierarchy.

Each managed folder has managed content settings associated with it. Managed content settings are a group of retention settings. However, a managed folder can have multiple managed content settings associated with it, so that Exchange can perform different actions on different types of content. These different types of content (such as e-mail messages, calendar items, and tasks) are known as message classes. Managed content settings can apply to a specific message class (such as applying only to calendar items), or it can apply to all message classes.

The retention settings for a managed folder include:

- The message class (or all message classes)
- The retention age (how long the specified message class should be retained)
- The retention action (what should happen to items once they reach their retention age)

Exchange 2010 provides five different retention actions that you can specify:

- **Move to Deleted Items Folder** – Managed content will be moved to the Deleted Items folder when it expires.
- **Move to Managed Custom Folder** – You can use this option to move expired content into a designated managed custom folder.
- **Delete and Allow Recovery** – This option moves expired content into the Recoverable Items folder, where the item can be recovered, if necessary, until the retention period specified at either the database or the mailbox expires. Keep in mind that the deleted item retention time is a disaster recovery feature, and is different from the retention action that is associated with managed folders.
- **Permanently Delete** – Expired content is deleted in a manner that prevents it from being recovered without restoring a backup.
- **Mark as Past Retention Limit** – When this action is specified, expired content is not deleted, but rather is displayed using strike through text. Strike through text is only supported by Outlook 2007 and above.

Although policies such as the ones that I have just described are relatively simple, there could potentially be a lot of work involved if you had to create a separate policy for each managed folder. As such, Managed Folder Mailbox Policies are designed in such a way that they can be applied to multiple folders. A managed folder mailbox policy can contain a mixture of default folders and custom folders, and a folder can be associated with a different policy at any time.

The Managed Folder Assistant

Another concept that you need to be familiar with is that of the Managed Folder Assistant. When a message expires, the specified retention action is not performed immediately. For example, suppose that a managed folder has a retention period of one day. If a message arrives in the folder at 10:43 a.m., it does not mean that the retention action will be performed at 10:43 a.m. the next day. Instead, the expired message will not be processed until later on when the Managed Folder Assistant runs.

The Managed Folder Assistant is a mechanism that runs on your mailbox servers from 1 to 9 a.m. each night. It is responsible for stamping mailbox items with retention ages as specified by the mailbox policy. The Managed Folder Assistant is also the mechanism that processes the retention action when items expire.

The Managed Folder Assistant can be very resource intensive, especially when it is run for the first time or when major policy changes are made. As such, it is important to schedule the Managed Folder Assistant to run at a time when any affected mailbox servers are not under a heavy load.

You can change the Manage Folder Assistant's schedule by using either the Exchange Management Console or the Exchange Management Shell. To use the Exchange Management Console to schedule the Managed Folder Assistant, navigate through the console tree to Server Configuration | Mailbox. Next, select the server schedule you want to manage from the result pane. Right click on your selected server, and choose the Properties command from the resulting shortcut menu.

At this point, Exchange will display the server's properties sheet. Go to the Message Records Management tab. Now, go to the Schedule Managed Folder Assistant section, and select the Use Custom Schedule option, and click the Customize button. You are now free to provide Exchange with a custom schedule. Click OK when you are done.

If you prefer to use the Exchange Management Shell, you can do so by using the Set-MailboxServer cmdlet. The command for doing so is:

```
Set-MailboxServer –Identity <server name>
-ManagedFolderAssistantSchedule "<schedule>"
```

For example, if you wanted to set the Managed Folder Assistant on a server named Domain1 to run from 10 to 11 p.m., you would use the following command:

```
Set-MailboxServer –Identity Domain1 –ManagedFolderAssistantSchedule
"Sun.22:00-Sun.23:00"
```

In some situations, you may find it useful to manually start the Managed Folder Assistant. For example, if you have just made some policy changes and want to process the changes right away, or if you are trying to troubleshoot a problem, then you probably will not want to have to wait for the Managed Folder Assistant's scheduled run time. In these types of situations, you can manually start the Managed Folder Assistant by entering the following command:

```
Start-ManagedFolderAssistant
```

When you launch the Managed Folder Assistant manually, it will continue to run until all objects have been processed. If you need to terminate the Managed Folder Assistant before it has completed, then you can do so by entering this command:

```
Stop-ManagedFolderAssistant
```

Retention Policies and Retention Tags

Now that you have been introduced to the concept of managed folders and the Managed Folder Assistant, it's time to discuss retention policies in greater detail.

The main thing that you need to understand about retention policies is that a retention policy is nothing more than a collection of individual retention tags. The retention tags are the mechanisms that Exchange uses to apply the individual policy elements that were discussed earlier. As such, it is critical that you understand what retention tags are and how they work.

Retention tags are used to stamp individual folders and messages with the corresponding retention requirements. There are three types of retention tags.

- **Default Policy Tag** – A retention policy can have one default policy tag. Exchange applies the Default Policy Tag to any item that does not already contain a retention tag. Although the Default Policy Tag is usually inherited, it can also be explicitly assigned.
- **Retention Policy Tags** – Retention Policy Tags are a folder level tag that is used by default folders.
- **Personal Tags** – Personal tags can be used to apply retention controls to custom folders or to individual mailbox items.

Exchange Server 2010 allows you to create any number of retention tags, but you are limited in the ways in which you can incorporate those tags into a retention policy. A retention policy is limited to using one Default Policy Tag. Likewise, a retention policy can only have one Retention Policy Tag per default folder. For example, if you wanted to apply retention settings to three different default folders, then the retention policy could have three Retention Policy Tags – one for each folder. Finally, a retention policy can contain an unlimited number of personal tags.

It is also worth noting that retention policies are applied to mailboxes. As such, you can create as many different retention policies as you need, but each mailbox is limited to using a single retention policy.

Creating Retention Tags

Although it is fairly easy to create retention tags, you do have to create them through the Exchange Management Shell. The cmdlet that you will use to do so is New-RetentionPolicyTag. There are several parameters that you must supply when you use this cmdlet:

- **The name of the tag** – When you create a new retention tag, you must assign it a name. There is not a parameter used for assigning the name. You simply specify the name within quotation marks immediately after the New-RetentionPolicyTag cmdlet.
- **Type** – You must tell Exchange what type of retention policy tag you are creating. If you are creating a default policy tag, then the type should be set to All. The Type for a Retention Policy Tag is set to the name of the default folder to which the tag will apply. If you are creating a Personal Tag, then the Type should be set to Personal.
- **Comment** – You should provide a comment to document the retention tag's purpose.
- **RetentionEnabled** – The RetentionEnabled parameter is used to express whether or not the retention tag should be enabled.
- **AgeLimitForRetention** – The AgeLimitForRetention parameter allows you to specify the age limit for items to which the tag applies.
- **RetentionAction** – The RetentionAction parameter allows you to control what should happen to items as they expire. Some of the options available to you include MoveToDeletedItems, MoveToArchive, and PermanentlyDelete.

Suppose, for instance, that you wanted to create a Retention Policy Tag for the deleted items folder that would cause items to be permanently deleted after 90 days. You could create such a tag by entering a command similar to this one:

```
New-RetentionPolicyTag "Tag-FIN-DeletedItems" -Type "DeletedItems" -
Comment "Items are removed after 90 days." -RetentionEnabled $True -
AgeLimitForRetention 90 -RetentionAction PermanentlyDelete
```

You can see what this command looks like in **Figure 25**.

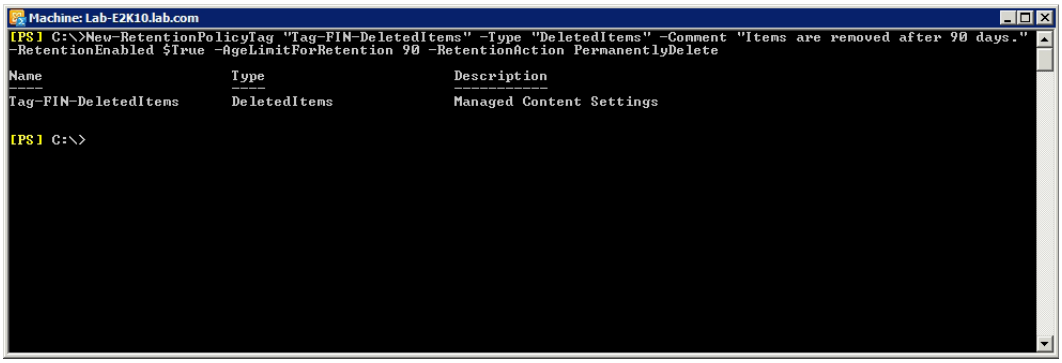

Figure 25: Creating a New Retention Policy Tag

This is what it looks like when you create a new retention policy tag.

Creating a Retention Policy

After you have created one or more retention tags, you can begin creating any necessary retention policies. As was the case with the retention tags, you will have to assign a name to each retention policy that you create. In addition to the policy name, you can specify the names of any tags that you want to associate with the policy. The name of each retention tag should be enclosed within quotation marks, and if you assign multiple retention tags, each tag should be separated by a comma.

In the previous section, we created a retention policy tag named Tag-FIN-DeletedItems. Suppose we wanted to create a retention policy named Domain1RetentionPolicy and assign our previously created retention tag to it. We could do so by using this command:

```
New-RetentionPolicy "MyDomain1RetentionPolicy" –
RetentionPolicyTagLinks "Tag-FIN-DeletedItems"
```

Figure 26 shows what it looks like when you execute this command.

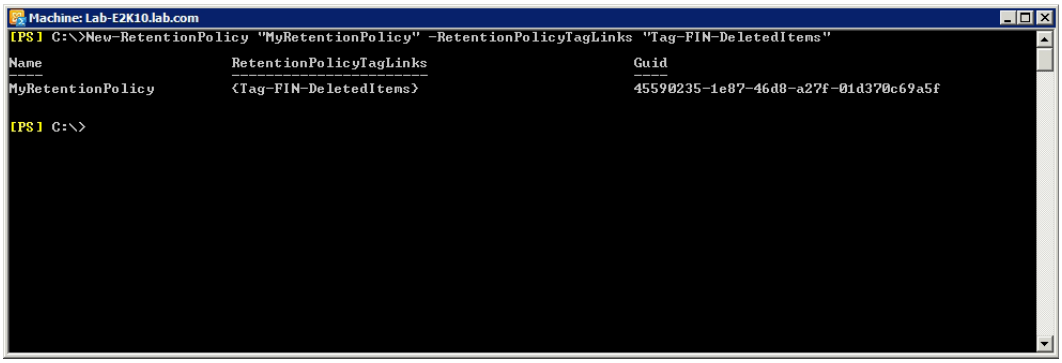

Figure 26: Showing a New Retention Policy

This is what it looks like when you create a retention policy.

Modifying a Retention Policy

Occasionally, you may find that you need to add some additional retention tags to an existing retention policy. You can accomplish this by using the Set-RetentionPolicy cmdlet, and providing the name of the retention policy and the names of the tags that you want to add to it. For example, suppose that we wanted to add a personal retention tag named BusinessCritical to the Domain1RetentionPolicy that we created in the previous section. To do so, we could use this command:

```
Set-RetentionPolicy "MyDomain1RetentionPolicy" –
RetentionPolicyTagLinks "Tag-FIN-DeletedItems","BusinessCritical"
```

You will notice in the command above that we included the name of the tag that was already in use in addition to the name of the new tag. The reason for doing so is that this command overwrites the previously existing retention policy. Had we omitted the previously used tag, the tag would no longer be included in the retention policy. This is how you can remove a tag from a retention policy.

Occasionally, a retention policy may be long and complex, and it may not be practical to retype the name of each tag that is associated with the policy. In such cases, Microsoft provides a set of commands that you can use to add a new tag to a retention policy without having to retype everything.

The following set of commands retrieves a list of the retention tags associated with the Domain1RetentionPolicy that we just created. Those tags are assigned to a variable named $TagList. The second line retrieves the settings for a brand new retention tag named Domain2Project, and assigns those settings to a variable named $NewTag. The $NewTag variable is then added to the $TagList variable so that it contains both the new and the previously existing retention tags. We wrap things up by using the $TagList variable within the Set-RetentionPolicy command. The actual commands are shown here:

```
$TagList = (Get-RetentionPolicy MyDomain1RetentionPolicy).
RetentionPolicyTagLinks
$NewTag = Get-RetentionPolicyTag Domain2Project
$TagList += $NewTag
Set-RetentionPolicy MyDomain1RetentionPolicy -RetentionPolicyTagLinks $TagList
```

Assigning a Retention Policy to a Mailbox

A retention policy doesn't do anything until you assign it to one or more mailboxes. You can assign a retention policy to a mailbox by using the Set-Mailbox cmdlet. For example, if you wanted to assign the Domain1RetentionPolicy retention policy that we created earlier to a mailbox belonging to a user with the account name JohnDoe, you could do so by using the following command:

```
Set-Mailbox "JohnDoe" –RetentionPolicy "MyDomain1RetentionPolicy"
```

Figure 27 shows an example of what happens when you apply a retention policy to a user's mailbox.

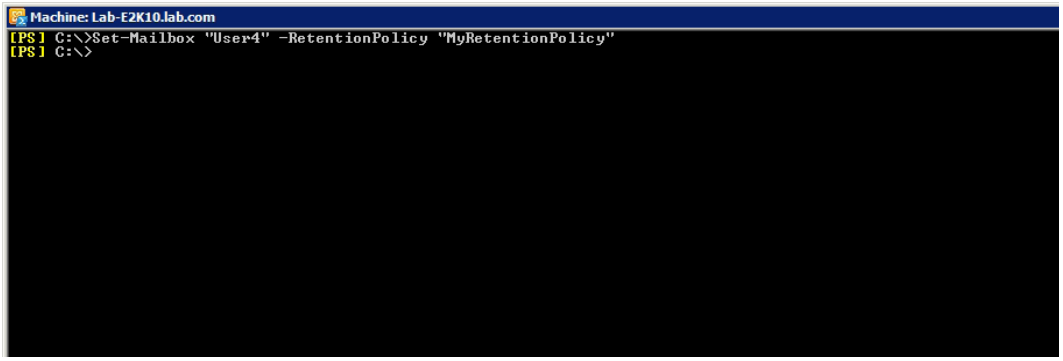

Figure 27: Applying a Retention Policy to a User's Mailbox

You can add a retention policy to a specific mailbox.

Of course in the real world, you would probably apply a retention policy on a per distribution group basis rather than applying it to individual mailboxes. You can apply a retention policy to a distribution group by using the Get-DistributionGroupMembers cmdlet to retrieve a list of the members of a distribution group. You would then channel pipe the output into the Set-Mailbox cmdlet as shown above. For example, if you wanted to assign the Domain1RetentionPolicy retention policy to members of the Marketing distribution group, you could do so by using the following command:

```
Get-DistributionGroupMember –Identity "Marketing" | Set-Mailbox –
RetentionPolicy "MyDomain1RetentionPolicy"
```

One thing that is critical for you to understand is that the command shown above only applies a retention policy to a distribution group's current members. If you remove a mailbox from a distribution group, any retention policies that were previously assigned to that mailbox through the distribution group will remain in effect. Likewise, if you add a mailbox to the distribution group, that mailbox will not automatically inherit the group's retention policy.

To get around this problem, you must run the command shown above any time that the group's membership changes. As an alternative, you configure your server to automatically run this command on a periodic basis according to a schedule.

Replacing a Retention Policy

Sometimes you may have to replace a retention policy with a new retention policy. For example, you may have to do this if your retention requirements for some users were to change. Although there is no straightforward way of doing this, you can use a series of Exchange Management Shell commands to get a list of all of the mailboxes to which a policy has been assigned, and then replace the old policy with the new policy.

For the sake of demonstration, suppose that some mailboxes in your organization had been assigned a retention policy named Domain1RetentionPolicy, and you wanted to change those mailboxes to use a retention policy named NewDomain1RetentionPolicy instead. You could do so by entering these commands:

```
$OldPolicy={Get-RetentionPolicy "Domain1RetentionPolicy"}.distinguishedName
Get-Mailbox -Filter {RetentionPolicy -eq $OldPolicy} -Resultsize
Unlimited | Set-Mailbox -RetentionPolicy "NewDomain1RetentionPolicy"
```

Retention Hold

One common problem with retention policies is that if a user takes a leave of absence, then messages received while the user is gone may expire before the user ever gets the chance to read them. Thankfully, Exchange administrators have the ability to place a retention hold on a mailbox as a way to keep this from happening. A retention hold keeps mailbox items from expiring until the hold is released.

You can use either the Exchange Management Console or the Exchange Management Shell to place a retention hold on a mailbox. To use the Exchange Management Console, navigate through the console tree to Recipient Configuration | Mailbox. In the result pane, right click on the mailbox that you want to place on hold, and then select the Properties command from the resulting shortcut menu. This will cause Exchange to display the mailbox's properties sheet.

Go to the properties sheet's Mailbox Settings tab, and then click on Message Records Management, followed by Properties. At this point, you will see a dialog box that you can use to enable the retention hold. Begin by selecting the Enable Retention Hold For Items In This Mailbox check box, and then specify the start and end date for the retention hold. Keep in mind that when the user does come back to work, it will probably take them a while to sort through all of their mail, so you may need to extend the retention hold beyond the user's return date so that the user will have time to work through the backlog.

Of course in some cases you may not know when the user is coming back. In those situations, it may be better to enable the retention hold without specifying any dates. To do so, enter the following command into the Exchange Management Shell:

```
Set-Mailbox "<user name>" -RetentionHoldEnabled $True
```

For example, to enable the retention hold for a user named John Doe, you could use this command:

```
Set-Mailbox "John Doe" -RetentionHoldEnabled $True
```

To release the retention hold, you would change the $True flag to $False, like this:

```
Set-Mailbox "John Doe" -RetentionHoldEnabled $False
```

Journaling

Many organizations are required by law to archive some or all of the messages that pass through their mail servers. Journaling is a feature that can make the archival process easier.

Archiving is technically defined as moving messages from a mail server into a safe place. Journaling does not archive messages in this sense, but rather makes a copy of each message and places it into a designated mailbox known as the journaling mailbox.

Exchange Server 2010 supports two different types of journaling:

- **Standard Journaling** – Standard Journaling is configured separately for each mailbox database. It works by journaling each message that is sent to or from mailboxes within the mailbox database.
- **Premium Journaling** – Premium journaling allows you to have more control over the journaling process by creating journal rules that control which messages (or which mailboxes) are journaled.

The reason why Exchange is able to journal messages is because the journal is actually a transport agent. In an Exchange 2010 organization, each message passes through a hub transport server, even if the message is destined for a mailbox residing within the same database as the mailbox that sent the message. The journal agent is able to intercept messages as they pass through the transport pipeline.

The journal rules themselves are stored in the Active Directory. As such, the rules are automatically replicated to each hub transport server within the organization.

One of the first things that you need to understand about journaling is the journal rule scope. Any time that you create a journal rule, you must assign a scope to that rule. Exchange 2010 supports three different scopes:

- **Internal** – The rule applies only to messages sent between your users.
- **External** – The rule applies to messages sent between your users and external recipients.
- **Global** – Journal rules with a global scope process all messages passing through a Hub Transport Server, even if those messages have already been processed by journal rules with an internal or external scope.

Another concept that you need to be familiar with is that of journal recipients. Simply put, a journal recipient is a user whose messages are being journaled. A journal recipient can be a user or a distribution group.

Journal Reports

When the journal agent journals a message, the Hub Transport Server does not simply forward the message to the journal mailbox. Instead, Exchange encapsulates the message within a journal report (which is sometimes referred to as the journal envelope).

The journal report is nothing more than an e-mail message containing the e-mail address of the original sender, the message's original recipient, the subject line, and the message ID. The original, unaltered message is provided as an attachment to the journal report.

Exchange uses journal reporting as a way of complying with the regulatory requirement that archived messages be unaltered.

Creating a Journal Mailbox

Although a journal mailbox acts as a repository for journaled messages, the mailbox is really nothing more than a dedicated user mailbox. Having said that, Microsoft advises that you take some steps to protect the mailbox's integrity.

One of the first steps that you should take is to ensure that the journal mailbox is only able to accept messages from the Microsoft Exchange Recipient. That way, users will not be able to submit fake journal reports to the journal mailbox.

There are two things that you need to know about locking down the journal mailbox in this way. First, you should not restrict the mailbox if journal entries are submitted by non Exchange mail servers. Second, the Microsoft Exchange recipient is a system mailbox and is not visible in the Global Address List. Therefore, you can only perform this procedure from the Exchange Management Shell.

To make it so that only the Microsoft Exchange recipient can deposit messages into the journal mailbox, use the following command:

```
Set-Mailbox "<journal mailbox name>" -AcceptMessagesOnlyFromSendersOrMembers
"Microsoft Exchange" -RequireSenderAuthenticationEnabled $True
```

You can see an example of this command in **Figure 28**.

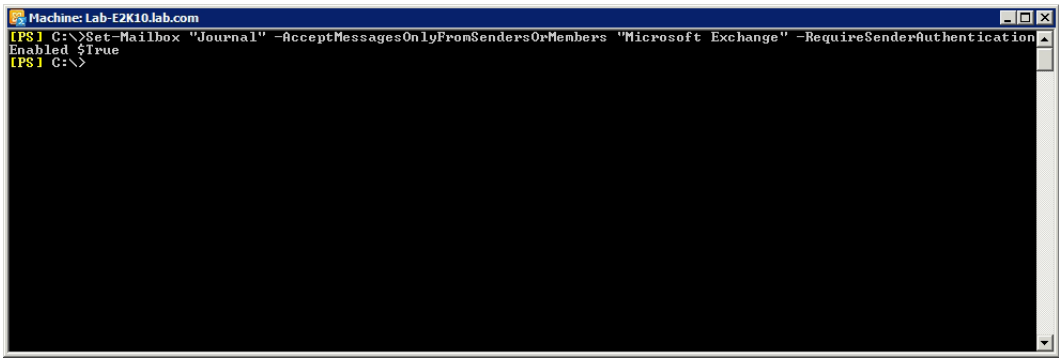

Figure 28: Configuring the Journal Mailbox

You should set the journal mailbox so that users cannot send messages to it.

The next thing that you should do is to make sure that no quota limits apply to the journal mailbox. That way, you won't have to worry about the mailbox filling up. You can disable any quota limits that may exist by using this command:

```
Set-Mailbox "<journal mailbox name>" -UseDatabaseQuotaDefaults
$false -IssueWarningQuota unlimited -ProhibitSendQuota unlimited
-ProhibitSendReceiveQuota unlimited
```

The last step is to give someone in your organization permission to open the journal mailbox. You can do so by using the Add-MailboxPermission cmdlet. For example, suppose that you wanted to give John Doe permission to open a journal mailbox named Journal. To do so, you would use the following command:

```
Add-MailboxPermission -Identity "Journal" -User JohnDoe -AccessRights
FullAccess -InheritanceType All
```

Creating a Journal Rule

Once you have created a journal mailbox, you can begin creating a journal rule. You can create a journal rule by using either the Exchange Management Console or the Exchange Management Shell.

To create a journal rule through the Exchange Management Console, navigate through the console tree to Organization Configuration | Hub Transport. Go to the Journal Rules tab found in the Results pane and click on the New Journal Rule link, located in the Action pane. When you do, Exchange will launch the New Journal Rule Wizard, shown in **Figure 29**.

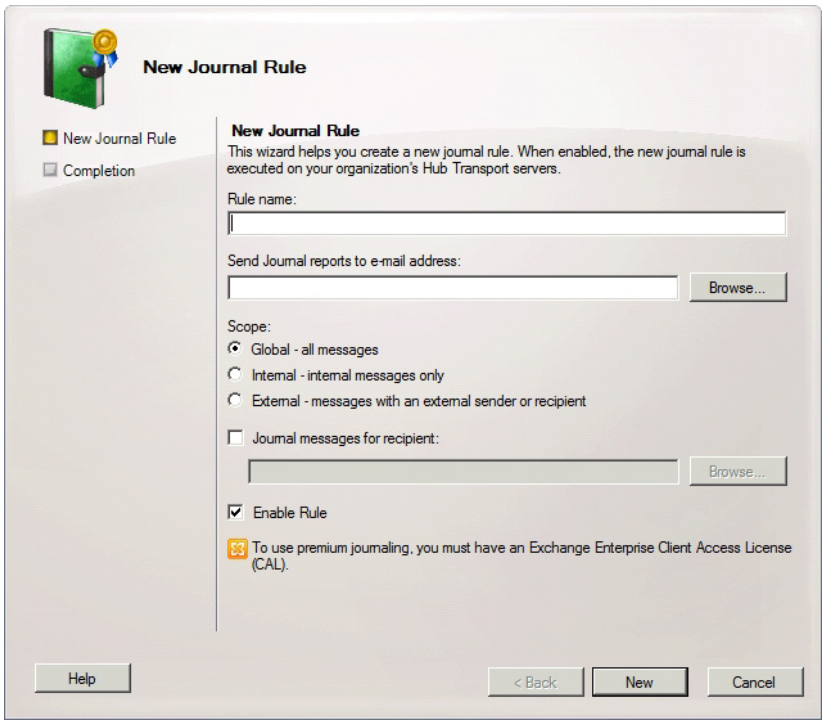

Figure 29: The New Journal Rule Wizard

The New Journal Rule Wizard is used for creating journal rules.

The wizard requires you to provide several pieces of information:

- **Rule Name** – This is the name that will be assigned to the journal rule that you are creating.
- **Send Journal Reports to E-Mail Address** – This is where you provide the name of the journal mailbox. Incidentally, Exchange will allow you to create multiple journal mailboxes, which you can use for varying purposes.
- **Scope** – This is where you set the rule's scope to Internal, External, or Global.
- **Journal Messages For Recipient** – You can use this option to journal all of the messages that are sent to or from a specific recipient. Alternatively, you can specify a distribution group rather than an individual mailbox.
- **Enable Rule** – This is a check box that causes the rule to be enabled once it is created. New journal rules are enabled by default.
- After populating the wizard's various fields, click New to create the rule. When Exchange confirms that the rule has been created, click Finish.

Just as you can use the Exchange Management Console to create a journal rule, you can also create journal rules by using the Exchange Management Shell. Doing so involves using the New-JournalRule cmdlet. This cmdlet requires you to provide the same information that you would use if you were using the wizard to create the new journal rule.

The syntax for the cmdlet is:

```
New-JournalRule -Name "<the name of the rule that you are creating>" -Recipients
<the e-mail address of the mailbox whose contents are being journaled>
-JournalEmailAddress "<the name of the journal mailbox>" -Scope <The rule scope
which can be internal, external, or global> -Enabled <$true or $false>
```

Here is an example of how this command might be used:

```
New-JournalRule -Name "Journal User1's Mail" -Recipient user1@Domain1.
com -JournalEmailAddress "Journal" -Scope Global -Enabled $True
```

Standard Journaling

The section above on creating journal rules provides examples of how to create journal rules when premium journaling is in use. Premium journaling is considered to be a premium feature that requires you to purchase a premium client access license. As such, it is also important that you know how to use standard journaling.

Standard journaling is considered to be a basic feature, and does not require a premium license. It is more limited than premium journaling because it doesn't allow you to journal individual mailboxes or distribution lists. Instead, standard journaling is enabled on a per database basis.

It may be worth noting that even though standard journaling is applied to individual databases, Microsoft considers it to be an organization level feature. This is because of the fact that a single database can reside on multiple mailbox servers, if database availability groups are in use.

Standard journaling can be implemented either through the Exchange Management Console or through the Exchange Management Shell. To use the Exchange Management Console, navigate through the console tree to Organization Configuration | Mailbox and then go to the Database Management tab, found in the results pane. Select the mailbox database on which you want to enable journaling, and then click the Properties link found in the Actions pane. When you do, Exchange will display a properties sheet for the database.

At this point, you must go to the properties sheet's Maintenance tab. Next, select the Journal Recipient check box and then click on the Browse button. Select the journal mailbox and then click OK.

If you prefer, you can enable standard journaling by using the Set-MailboxDatabase cmdlet. The syntax for the command that you must use is as follows:

```
Set-MailboxDatabase "<database name>" -JournalRecipient "<journal mailbox name>"
```

For example, suppose that you wanted to enable journaling for a mailbox database named DB1, and you wanted to send the journal reports to a journal mailbox named Journal. To do so, you would use the following command:

```
Set-MailboxDatabase "DB1" -JournalRecipient "Journal"
```

You may recall from the section on Database Availability Groups that it is sometimes necessary to specify which server contains the active copy of a database when multiple replicas exist. Even though standard journaling fully supports database availability groups, it is server independent. Therefore, you can specify the database name without having to worry about providing the server name.

Message Classification

Many organizations are subject to government regulations that stipulate how various types of e-mail messages must be handled. Microsoft provides message classifications as a way of making it easier to ensure that messages are handled properly.

The act of classifying a message appends metadata to the message that provides instructions on what the message's intended purpose is and how the message should be handled.

By default, message classifications don't actually do anything other than causing Outlook or Outlook Web App to display a banner stating how the message has been classified. That way, the recipient knows what they are and are not allowed to do with the message. For example, if a user receives a message that is classified as Confidential, then the user knows that the message should not be forwarded to someone outside of the organization.

Even though message classifications are initially configured to be informational only, they don't have to be. In fact, most of the organizations that use message classifications do so in conjunction with transport rules.

Transport rules can be configured to automatically classify a message, or they can be configured to act on classifications that were put in place by end users.

One example of having a transport rule classify a message would be a situation in which the HR department wanted to ensure that nobody is using e-mail for abusive purposes. In doing so, an organization might create a transport rule that checks messages to see if they contain swear words. Messages containing such words might be classified as policy violations and then rerouted to a designated mailbox.

Likewise, transport rules can act on messages that are classified by end users. For example, a transport rule might look for messages that have been classified as confidential, and then archive a copy of the message to a designated mailbox.

It is worth noting that classifications such as Confidential and Policy Violation do not exist by default, but Exchange does allow you to create custom classifications on an as needed basis. In fact, there are only three message classifications enabled in Exchange 2010 by default:

- **Attachment Removed** – This classification notifies a message recipient that an attachment has been stripped from the message.
- **Originator Requested Alternate Recipient Mail** – The Originator Requested Alternate Recipient Mail classification tells the recipient that the message has been redirected and was not delivered to the originally addressed recipient.
- **Partner Mail** – This classification tells the recipient that the message was encrypted and sent through a secure connector.

It is worth noting that the default message classifications can only be added to a message by Exchange Server. They cannot be added to a message by an end user.

Creating a New Message Classification

Exchange Server 2010 does not limit you to using only the built in message classifications. You can easily create your own custom message classifications that can be used alongside the default classifications. Doing so involves using the New-MessageClassification cmdlet.

When you create a new message classification, there are several key pieces of information that you must provide:

- **Display Name** – The display name is the message classification's friendly name.
- **Locale** – The locale is a localization code indicating the classification's language. The locale code used for English is en-EN.
- **Name** – This is the name that is assigned to the classification when it is created. You will use this name any time that you have to use the Exchange Management Shell to perform an action against the message classification.
- **Sender Description** – The sender description is arguably the most important part of the classification. It is the text that tells the sender and the recipient how the message should be handled.

To see how creating a new message classification works, imagine that you want to create a classification that can be used to flag a message as containing text that has been approved for release to the media. We will call this new message classification Media. The command for creating such a classification would look something like this:

```
New-MessageClassification –Name Media –DisplayName "Media Release" –
SenderDescription "The contents of this message have been approved for
release to the media." –Locale en-EN
```

Configuring Outlook to Support Message Classifications

As strange as it may seem, Outlook does not support message classifications by default (although Outlook Web App does). If you want to allow users to classify messages or to view message classifications, you must configure Outlook to support classifications.

The first step in configuring Outlook to support message classifications is to create an XML file containing a list of all of the message classifications that have been defined within Exchange. Thankfully, Microsoft provides a script that you can use to create the necessary XML file.

The script is called Export-OutlookClassification.ps1, and is located in the Exchange Server's \Program Files\Microsoft\Exchange Server\V14\Scripts folder. When you run the script, you must redirect the output to an XML file. For instance, you could use the command shown below to create a file named C:\Classifications.xml.

```
./Export-OutlookClassification.ps1 > C:\Classifications.xml
```

There are a couple of important things that you need to keep in mind about this XML file. First, there is a chance that it could contain message classifications that you do not want to expose through Outlook. As such, it is a good idea to manually edit the file and remove any references to message classifications that you do not want your users to use.

Another thing to keep in mind about this file is that it will not be updated automatically. As such, you should take care to make sure that all of the necessary classifications have been created before you generate the XML file. Otherwise, you will have to recreate and redistribute the XML file any time that you make a change to the message classifications defined on the Exchange Server.

After you create the XML file, you must distribute it so that each copy of Outlook has access to the file. Technically, it is possible to place the XML file onto a network share. However, Microsoft discourages doing so because users operating in cache mode will only have access to classification information when they are connected to Exchange. This can be especially problematic for mobile users.

Once you have distributed the XML file, you must create a registry key and a series of registry settings on each computer that will be running Outlook. As you do, it is important to remember that editing the registry is dangerous. If you make a mistake while modifying the registry, you can destroy Windows and/or your applications. As such, you should make a backup prior to making any registry modification. That being said, there are three registry settings that you will have to provide.

The first setting is AdminClassificationPath. This registry setting references the path to the Classification.xml file.

The second setting is EnableClassifications. As the name implies, this is the setting that enables or disables message classification support for Outlook. A DWORD value of 00000001 enables message classifications, while a value of 00000000 disables message classification support.

The third setting is TrustClassifications. This setting, which should only be enabled for users whose mailboxes are stored on Exchange 2010 servers, is based on the fact that Exchange 2003 does not support message classifications. Enabling this setting by assigning it a value of 00000001 causes the words "The Sender Claims" to be added to the message classification as a way of alerting the users that the message's classification cannot be verified.

To create the necessary registry settings, you must begin by creating a registry key at:

```
HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Common\Policy
```

After doing so, you should add the following settings beneath the key:

```
AdminClassificationPath
String Value
c:\\Classifications.xml

EnableClassifications
DWORD value
00000001

TrustClassifications
DWORD value
00000001
```