

# 7

## Configuring the Hub Transport Role

Exchange Server 2000/2003 used the concept of bridgehead servers and connectors. A bridgehead server referred to an Exchange server that served as a connection point for delivering email from one routing group to another and to remote or external email systems. Bridgehead servers used connectors to make information flow between routing groups and remote or external systems possible. Several types of connectors were available: SMTP, Routing Group, and X.400.

Exchange Server 2007 introduces the concept of the Hub Transport role. Computers running Exchange Server 2007 with the Hub Transport role are called Hub Transport servers and are identical to bridgehead servers in Exchange 2000/2003; however, they differ greatly in core transport functionality. The Hub Transport server role is installed in any Active Directory site that contains the Mailbox server role and is responsible for mail delivery within the Active Directory site. It can be installed on separate hardware as the only server role or on the same server hardware in conjunction with other non-clustered Exchange Server 2007 roles. The Hub Transport server receives messages from and sends messages to servers running the Mailbox server role. Every message sent and received by an Exchange mailbox must pass through the Hub Transport server, hence transport rules and journal policies are not skipped for any message. In a multi-site organization, messages destined for a user in a different site are transferred to a Hub Transport server in that site for delivery. Messages destined for the Internet or other messaging systems are sent to the Edge Transport server for delivery. We discuss the Edge Transport role further in Chapter 9. The Hub Transport server role uses *Send* Connectors and *Receive* Connectors for email routing and delivery.

This chapter covers:

- Understanding the core transport architecture implemented by the Hub Transport and Edge Transport servers.
- Using and configuring the Hub Transport server
- Configuring various types of connectors in Exchange Server 2007
- Using email address policies and accepted domains

# The Transport Server Architecture

The core transport architecture was rewritten in Exchange Server 2007 and is very different from previous versions of Exchange. Those familiar with Exchange Server 2000/2003 might quickly notice that transport is no longer dependent on Internet Information Server (IIS). In fact, it is required that you uninstall the SMTP and NNTP services prior to installing Exchange Server 2007 unlike Exchange Server 2000/2003, which required both services to be installed. Additionally, all core components required for message categorization, routing, and delivery are included in Exchange Transport Service with no components dependent on IIS. This section briefly reviews the core transport architecture from the perspective of the Management Shell.

The following Hub Transport–related cmdlets are discussed:

- `Get-Queue`
- `Set-Queue`
- `Suspend-Queue`
- `Resume-Queue`
- `Retry-Queue`
- `Get-TransportPipeline`
- `Get-TransportServer`
- `Set-TransportServer`
- `Get-TransportConfig`
- `Set-TransportConfig`
- `Get-NetworkConnectionInfo`

A number of components make up the core transport architecture implemented by both the Hub Transport server and Edge Transport server roles. These components as well as other processes and queues constitute the *transport pipeline* in Exchange Server 2007. Think of it as a series of processes that make message delivery or relay possible. Every message sent or received must go through the transport pipeline. The transport pipeline consists of the following:

- SMTP Receive:** This component accepts connections on port 25 inbound to the Hub Transport or Edge Transport servers. This component is controlled by the SMTP Receive Connector, which is similar to the SMTP virtual server in Exchange Server 2000/2003. It is at this stage of the transport pipeline that anti-virus and anti-spam agents are implemented to filter incoming connections, message content, determine the sender, and apply any compliance or transport rules configured. Actual message hygiene or transport rules performed vary slightly depending on which server role is installed, either the Hub Transport role or Edge Transport role. A series of events are triggered as the message is received and agents are executed against the message.

## Chapter 7: Configuring the Hub Transport Role

- ❑ **Submission Queue:** After a message is accepted into the organization either from SMTP Receive or Pickup/Replay directory, it is placed into the Submission queue by the Submission process. Submission can also occur when the store driver retrieves outbound messages from users' outboxes and places them in the Submission queue. (See Figure 7-1.) This queue is essentially an ESE database similar to the server mailbox store database. This differs markedly from Exchange Server 2000/2003 where incoming SMTP messages or messages from the pickup queue were placed in the Queue folder, a physical NTFS partition.

```
Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-Queue

Identity                DeliveryType Status MessageCount NextHopDomain
-----
HT001\Submission        Undefined   Ready    0              Submission

[PS] C:\>Get-Queue | fl

Identity                : HT001\Submission
DeliveryType            : Undefined
NextHopDomain           : Submission
NextHopConnector        : 00000000-0000-0000-0000-000000000000
Status                  : Ready
MessageCount            : 0
LastError               :
LastRetryTime           :
NextRetryTime           :
IsValid                 : True
ObjectState             : Unchanged

[PS] C:\>Get-Queue sub* | Suspend-Queue

Confirm
Are you sure you want to perform this action?
Suspending the queue "HT001\Submission".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help
(default is "Y"):a
[PS] C:\>Get-Queue sub*

Identity                DeliveryType Status MessageCount NextHopDomain
-----
HT001\Submission        Undefined   Sus... 0              Submission

[PS] C:\>Get-Queue sub* | Resume-Queue
[PS] C:\>Get-Queue sub*

Identity                DeliveryType Status MessageCount NextHopDomain
-----
HT001\Submission        Undefined   Ready    0              Submission

[PS] C:\>_
```

Figure 7-1

## Part II: Working with Server Roles

As shown in Figure 7-1, when there are no messages sent or delivered by the Hub Transport the `Get-Queue` cmdlet returns only the Submission queue. Notice that the `DeliveryType` is undefined and `NextHopConnector` is all zeros. The next hop is the Categorizer. Like any other message queue in Exchange Server 2007, it can be suspended and resumed. When this is done, the Hub Transport server no longer processes new incoming messages; rather, the message count continually increases by the number of new messages received. When the queue is resumed, the Submission queue is de-queued and messages are picked up once again by the Categorizer. The `Suspend-Queue` and `Resume-Queue` cmdlets are used to pause and resume message queues as shown in Figure 7-1. Notice the change in queue status when the queue is suspended and when resumed. With the exception of the Submission queue, for other message queues, the `Retry-Queue` cmdlet can be used to retry messages in the queue after a transient failure.

- ❑ **Categorizer:** The Categorizer collects and processes messages placed in the Submission queue. This key component of the transport pipeline is responsible for several functions depending on whether the transport server is the Hub or Edge server role. On the Edge server, categorization simply involves routing the submitted message to a delivery queue based on the recipient domain. On the Hub Transport server, categorization involves recipient resolution, distribution list expansion, message content conversion, routing, and application of any rules defined. Thereafter message delivery is either MAPI Delivery or Remote Delivery to another Hub server or Edge Transport server. On the Edge Transport server, unlike the Hub Transport server, after categorization, message delivery will always be via Send Connectors to the target destination. Figure 7-4 shows the transport pipeline exposing two events of the Categorizer: the `OnSubmittedMessage` and `OnRoutedMessage` events.
- ❑ **Local (MAPI) Delivery:** This stage of the transport pipeline delivers messages from a Hub Transport server to a mailbox on a mailbox server in the Active Directory site. After a message has been categorized and its next hop identified as a mailbox store within the Active Directory site, the message is moved to the MAPI Delivery queue. The store driver component involved at this stage connects to the Recipients mailbox store and writes the message to the inbox, after which the message is deleted from the MAPI Delivery queue. There can be multiple MAPI Delivery queues depending on the number of mailbox servers in the local site for which messages are destined. Figure 7-2 shows local delivery queues to multiple mailbox servers when the `Get-Queue` cmdlet is run.
- ❑ **SMTP Send/Remote Delivery:** After categorization, messages destined for users not in the local Active Directory site or for remote SMTP servers or domains are placed in the Remote Delivery queue. This component is controlled by the SMTP Send Connector, which is similar to the SMTP Connector in Exchange Server 2000/2003. There can be several Remote Delivery queues and you may see a separate queue for each remote domain that messages are to be delivered to. If an Edge Transport server exists, remote delivery for all Internet domains will be through the Send Connector to the Edge Transport server. If coexisting with an earlier version of Exchange, messages destined for these servers will be relayed to the routing group where these servers reside. Figure 7-3 shows two Remote Delivery queues, one with a `deliverytype` of `SmtprelayToRemoteAdSite` to a Hub server in a different Active Directory site and another with `Deliverytype` called `SmtprelayToTiRg` with the next hop being an Exchange 2003 server in the “First Routing” group. The `MAPIDelivery` queue shown is discussed shortly.

## Chapter 7: Configuring the Hub Transport Role

```

Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-Queue

Identity                DeliveryType  Status  MessageCount  NextHopDomain
-----
HT001\5                 MapiDelivery Ready    0              gk-hcm.exchangeexch...
HT001\6                 MapiDelivery Ready    0              ht001.exchangeexcha...
HT001\Submission       Undefined     Ready    0              Submission

[PS] C:\>
[PS] C:\>
[PS] C:\>Get-Queue | fl

Identity                : HT001\5
DeliveryType            : MapiDelivery
NextHopDomain           : gk-hcm.exchangeexchange.local
NextHopConnector        : 00000000-0000-0000-0000-000000000000
Status                  : Ready
MessageCount            : 0
LastError               :
LastRetryTime           : 10/5/2007 12:28:54 AM
NextRetryTime           :
IsValid                 : True
ObjectState             : Unchanged

Identity                : HT001\6
DeliveryType            : MapiDelivery
NextHopDomain           : ht001.exchangeexchange.local
NextHopConnector        : 00000000-0000-0000-0000-000000000000
Status                  : Ready
MessageCount            : 0
LastError               :
LastRetryTime           : 10/5/2007 12:28:54 AM
NextRetryTime           :
IsValid                 : True
ObjectState             : Unchanged

Identity                : HT001\Submission
DeliveryType            : Undefined
NextHopDomain           : Submission
NextHopConnector        : 00000000-0000-0000-0000-000000000000
Status                  : Ready
MessageCount            : 0
LastError               :
LastRetryTime           :
NextRetryTime           :
IsValid                 : True
ObjectState             : Unchanged

[PS] C:\>_

```

Figure 7-2

```

Professional PowerShell for Exchange 2007 | Scope: View Entire Forest
[PS] C:\>get-queue | ft -autosize

Identity                DeliveryType  Status  MessageCount  NextHopDomain
-----
GCEX-7A\15              SmtRelayToTiRg  Ready    0              cn=first rout...
GCEX-7A\16              MapiDelivery   Ready    0              gcex-7a.afr.g...
GCEX-7A\17              SmtRelayToRemoteAdSite  Ready    0              default-first...
GCEX-7A\Submission     Undefined     Ready    0              Submission

[PS] C:\>_

```

Figure 7-3



## Chapter 7: Configuring the Hub Transport Role

another Active Directory site, or to an Edge Transport server for Internet delivery. The message can also be routed directly from the Hub Transport server if no Edge Transport server is configured.

### **Configuring the Hub Transport Server**

The Hub Transport server implements the core transport functionality and is responsible for all message flow within an Exchange organization. As mentioned earlier, it must be deployed into any Active Directory site that contains the Mailbox server role. All configuration information for the Hub Transport server is stored in Active Directory and changes made take effect on all Hub Transport servers in the organization. By default, the Hub Transport server configures and enables two transport policy and compliance agents, the Transport Rule Agent and the Journaling Agent. Unified Messaging messages such as voice and fax messages could bypass transport rules; however, in Exchange Server 2007 SP1 transport rules now act on Unified Messaging messages.

The `Get-TransportServer` and `Set-TransportServer` cmdlets enable you to view and change the property configuration on the Hub Transport server. Changing the configuration alters how the server processes messages. These changes are not organization wide, but affect only the specified server. This differs from the `Get-TransportConfig` and `Set-TransportConfig` cmdlets, which enable you to view and modify transport configuration for the whole Microsoft Exchange Server 2007 organization.

The `Get-TransportServer` cmdlet displays transport information for computers running the Hub Transport or Edge Transport role in the Exchange organization. It has two parameters: `Identity` for specifying the Hub Transport server to retrieve its information and `DomainController`, the fully qualified domain name of the domain controller used to retrieve the Hub Transport's server information. These parameters do not apply to the Edge Transport server. The `Identity` parameter always returns the local Edge Transport server and the domain controller parameter is not supported on the Edge Transport server because it is installed in a perimeter network with no access to Active Directory on a domain controller. Figures 7-5 and 7-6 display all the transport server settings available on the Hub Transport server.

The following section takes a quick look at a couple of the settings.

### **DNS Configuration**

By default if `InternalDNSAdapterEnabled` is set to `True`, the Hub Transport server will always use the DNS servers configured on the internal network adapter. This is also the case even if `ExternalDNSAdapterEnabled` is `True`. If `InternalDNSAdapterEnabled` is set to `False` and DNS servers are manually specified, then the servers listed in the `InternalDNSServers` parameter will always be used. The Hub Transport only uses DNS servers specified with the `ExternalDNSServers` parameter or the external network adapter when a Send Connector is configured to use it.

In Figure 7-5, notice that both `InternalDNSAdapterEnabled` and `ExternalDNSAdapterEnabled` parameters are set to `True` on the transport server. This is the case even if the server is configured with a single network card. The DNS server(s) specified on the network card is used. Also, both `InternalDNSAdapterGuid` and `ExternalDNSAdapterGuid` are all zeros, indicating that DNS lookups will be performed using any available adapter on the server. To use a list of DNS servers other than those on the adapter, first disable `InternalDNSAdapterEnabled` and `ExternalDNSAdapterEnabled` by setting them to `false` using the `Set-TransportServer` cmdlet, then specify the DNS servers using the `ExternalDNSServers` or `InternalDNSServers` parameter. If multiple adapters exist and you

## Part II: Working with Server Roles

```
Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-TransportServer HT001 | fl

Name : HT001
AntispamAgentsEnabled : False
ConnectivityLogEnabled : False
ConnectivityLogMaxAge : 30.00:00:00
ConnectivityLogMaxDirectorySize : 250MB
ConnectivityLogMaxFileSize : 10MB
ConnectivityLogPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\Connectivit
y
DelayNotificationTimeout : 04:00:00
ExternalDelayDsnEnabled : True
ExternalDNSAdapterEnabled : True
ExternalDNSAdapterGuid : 00000000-0000-0000-0000-000000000000
ExternalDNSProtocolOption : Any
ExternalDNSServers : {}
ExternalIPAddress :
ExternalDsnDefaultLanguage : en-US
ExternalDsnLanguageDetectionEnabled : True
ExternalDsnMaxMessageAttachSize : 10MB
ExternalDsnReportingAuthority : ExchangeExchange.local
ExternalDsnSendHtml : True
ExternalPostmasterAddress :
InternalDelayDsnEnabled : True
InternalDNSAdapterEnabled : True
InternalDNSAdapterGuid : 00000000-0000-0000-0000-000000000000
InternalDNSProtocolOption : Any
InternalDNSServers : {}
InternalDsnDefaultLanguage : en-US
InternalDsnLanguageDetectionEnabled : True
InternalDsnMaxMessageAttachSize : 10MB
InternalDsnReportingAuthority : HT001.ExchangeExchange.local
InternalDsnSendHtml : True
MaxConcurrentMailboxDeliveries : 7
MaxConcurrentMailboxSubmissions : 20
MaxConnectionRatePerMinute : 1200
MaxOutboundConnections : 1000
MaxPerDomainOutboundConnections : 20
MessageExpirationTimeout : 2.00:00:00
MessageRetryInterval : 00:01:00
MessageTrackingLogEnabled : True
MessageTrackingLogMaxAge : 30.00:00:00
MessageTrackingLogMaxDirectorySize : 250MB
MessageTrackingLogMaxFileSize : 10MB
MessageTrackingLogPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\MessageTrac
king
MessageTrackingLogSubjectLoggingEnabled : True
OutboundConnectionFailureRetryInterval : 00:10:00
IntraOrgConnectorProtocolLoggingLevel : None
PickupDirectoryMaxHeaderSize : 64KB
PickupDirectoryMaxMessagesPerMinute : 100
PickupDirectoryMaxRecipientsPerMessage : 100
PickupDirectoryPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Pickup
PipelineTracingEnabled : False
ContentConversionTracingEnabled : False
PipelineTracingPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\PipelineTra
cing
PipelineTracingSenderAddress :
PoisonMessageDetectionEnabled : True
```

Figure 7-5

```

PoisonThreshold : 2
QueueMaxIdleTime : 00:03:00
ReceiveProtocolLogMaxAge : 30.00:00:00
ReceiveProtocolLogMaxDirectorySize : 250MB
ReceiveProtocolLogMaxFileSize : 10MB
ReceiveProtocolLogPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\ProtocolLog
\SmtptReceive

RecipientValidationCacheEnabled : False
ReplayDirectoryPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Replay

RootDropDirectoryPath :
RoutingTableLogMaxAge : 7.00:00:00
RoutingTableLogMaxDirectorySize : 50MB
RoutingTableLogPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\Routing

SendProtocolLogMaxAge : 30.00:00:00
SendProtocolLogMaxDirectorySize : 250MB
SendProtocolLogMaxFileSize : 10MB
SendProtocolLogPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\ProtocolLog
\SmtptSend

TransientFailureRetryCount : 6
TransientFailureRetryInterval : 00:05:00
AntispamUpdatesEnabled : False
IsValid : True
OriginatingServer : GK-GC.ExchangeExchange.local
ExchangeVersion : 0.1 (8.0.535.0)
DistinguishedName : CN=HT001,CN=Servers,CN=Exchange Admin
istrative Group (FYDIBOHF23SPDLT),CN=
Administrative Groups,CN=ExchangeExch
ange,CN=Microsoft Exchange,CN=Service
s,CN=Configuration,DC=ExchangeExchang
e,DC=local

Identity : HT001
Guid : 09d9f0a8-e928-4caf-82eb-5fc0875d3afe
ObjectCategory : ExchangeExchange.local/Configuration/
Schema/ms-Exch-Exchange-Server

ObjectClass : <top,server,msExchExchangeServer>
WhenChanged : 7/2/2007 9:55:36 PM
WhenCreated : 7/2/2007 9:43:28 PM

[PS] C:\>

```

Figure 7-6

intend to use a specific adapter, specify the adapter by its GUID. The GUID can be obtained using the `Get-NetworkConnectionInfo` cmdlet.

Figure 7-7 shows how to configure the transport server to use a specific adapter for internal and external lookup by specifying its GUID and view changes made.

## Back Pressure

Hub Transport and Edge Transport servers monitor important system resources using the back pressure feature. System resources monitored include available disk space and memory. This feature enables transport servers to reject new connections and messages when a configured resource threshold is reached. In the RTM release of Exchange Server 2007, the disk resource limit was 4GB, hence if a Hub Transport server has less than 4GB of drive space all new mail processing stops. However, this has been changed to 500MB in Exchange Server 2007 SP1. The configuration options for back pressure can be modified in the `EdgeTransport.exe.config` application configuration file.

```

Select Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>$net = Get-NetworkConnectionInfo
[PS] C:\>$net

Name           : Microsoft UMBus Network Adapter #2
DnsServers     : <172.16.8.51>
IPAddresses    : <172.16.8.53>
AdapterGuid    : 41e55f5c-c8df-432b-85cb-daf967ba3536
MacAddress     : 00:15:5D:44:47:0B

[PS] C:\>$guid = $net.AdapterGuid
[PS] C:\>Set-TransportServer HT001 -InternalDNSAdapterGuid $guid -ExternalDNSAd
[PS] C:\>Get-TransportServer HT001 | fl name, *DNS*

Name           : HT001
ExternalDNSAdapterEnabled : True
ExternalDNSAdapterGuid   : 41e55f5c-c8df-432b-85cb-daf967ba3536
ExternalDNSProtocolOption : Any
ExternalDNSServers       : <>
InternalDNSAdapterEnabled : True
InternalDNSAdapterGuid   : 41e55f5c-c8df-432b-85cb-daf967ba3536
InternalDNSProtocolOption : Any
InternalDNSServers       : <>

[PS] C:\>_
    
```

Figure 7-7

### Priority Queuing

Priority queuing is new to Exchange Server 2007 SP1 and allows Hub Transport servers to process messages based on the priority defined by the sender (Low, Normal, High). As discussed earlier in this chapter, after a message is categorized, it is placed in a delivery queue, either local (MAPI) or remote delivery. With the Priority Queuing feature enabled, all messages placed in a specific delivery queue are processed to their destination based on the message priority stored in the X-Priority header field. As with the back pressure feature, the configuration options priority queuing can be modified in the `EdgeTransport.exe.config` application configuration file. However, the settings configured override the limits set by the `Set-TransportServer` cmdlet (discussed in the following section). Figure 7-8 shows timeout values configured on the Hub server.

```

Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-TransportServer HT001 | fl name, *timeout*

Name           : HT001
DelayNotificationTimeout : 04:00:00
MessageExpirationTimeout : 02:00:00

[PS] C:\>_
    
```

Figure 7-8

## Transport Server Limits

Several limits that apply to message and connection retry attempts, message expiration, connection limits, and restrictions can be set on the Hub server. Using the `Set-TransportServer` cmdlet you can modify these settings. For example, when a message delivery fails due to a transient failure, it continually retries and expires after two days. You can change the `MessageExpirationTimeout` value within the range of 1 and 90 days. The same applies to the transport log settings for the message tracking, connectivity, and Protocol logs. Figure 7-9 shows how to modify message expiration timeout settings.

```

[PS] C:\>Set-TransportServer HT001 -MessageExpirationTimeout 10:00:00
[PS] C:\>Get-TransportServer HT001 | fl

Name : HT001
AntispamAgentsEnabled : False
ConnectivityLogEnabled : False
ConnectivityLogMaxAge : 30.00:00:00
ConnectivityLogMaxDirectorySize : 250MB
ConnectivityLogMaxFileSize : 10MB
ConnectivityLogPath : C:\Program Files\Microsoft\Exchange S
erver\TransportRoles\Logs\Connectivit
y
DelayNotificationTimeout : 04:00:00
ExternalDelayDsnEnabled : True
ExternalDNSAdapterEnabled : True
ExternalDNSAdapterGuid : 41e55f5c-c8df-432b-85cb-daf967ba3536
ExternalDNSProtocolOption : Any
ExternalDNSServers : <>
ExternalIPAddress :
ExternalDsnDefaultLanguage : en-US
ExternalDsnLanguageDetectionEnabled : True
ExternalDsnMaxMessageAttachSize : 10MB
ExternalDsnReportingAuthority : ExchangeExchange.local
ExternalDsnSendHtml : True
ExternalPostmasterAddress :
InternalDelayDsnEnabled : True
InternalDNSAdapterEnabled : True
InternalDNSAdapterGuid : 41e55f5c-c8df-432b-85cb-daf967ba3536
InternalDNSProtocolOption : Any
InternalDNSServers : <>
InternalDsnDefaultLanguage : en-US
InternalDsnLanguageDetectionEnabled : True
InternalDsnMaxMessageAttachSize : 10MB
InternalDsnReportingAuthority : HT001.ExchangeExchange.local
InternalDsnSendHtml : True
MaxConcurrentMailboxDeliveries : 7
MaxConcurrentMailboxSubmissions : 20
MaxConnectionRatePerMinute : 1200
MaxOutboundConnections : 1000
MaxPerDomainOutboundConnections : 20
MessageExpirationTimeout : 10:00:00
MessageRetryInterval : 00:01:00
  
```

Figure 7-9

## Creating and Modifying Connectors

Connectors determine the path a message takes when routed between servers within an organization, to and from the Internet, and between messaging organizations. They provide single direction or one-way connections between a source server and a destination server. In Exchange Server 2007, at least four types of connectors can be explicitly created and configured. These include the Send Connector, Receive Connector, Routing Group Connector, and Foreign Connector. There also exists implicitly created Send

## Part II: Working with Server Roles

---

or Receive Connectors created on Hub Transport server's internal mail flow within the Active Directory site and across Active Directory sites in an Exchange Server 2007 forest. This enables Hub Transport servers to communicate with each other. Hence, you do not have to configure any connectors between Hub Transport servers within the Active Directory forest. If an Edge Transport server exists in the organization, the Edge Subscription process is recommended to automatically create and configure connectors between the Edge Transport server and a Hub Transport server in a designated Active Directory site. The following cmdlets are covered in this section; some are discussed later in more detail.

- `Get-SendConnector`
- `Set-SendConnector`
- `New-SendConnector`
- `Remove-SendConnector`
- `Get-ReceiveConnector`
- `Set-ReceiveConnector`
- `New-ReceiveConnector`
- `Remove-SendConnector`
- `Format-List`
- `Get-TransportConfig`
- `Set-TransportConfig`
- `Get-ADPermission`

### ***New-ReceiveConnector***

The `New-ReceiveConnector` cmdlet creates a new Receive Connector. The `Name`, `Bindings`, and, `RemoteIPRanges` are required parameters. The following switch parameters are important to note: `Custom`, `Client`, `Partner`, `Internal`, and `Internet`. These indicate the types of Receive Connectors that can be created. They can be specified separately or using the `Usage` parameter but cannot be used in conjunction with this parameter. If no usage type is specified, the default usage type of custom will be used.

- `Custom <SwitchParameter>`: This parameter can be used to specify the custom usage type. The usage type specifies the default permission groups and authentication methods that are assigned to this connector. The Custom Receive Connector is a customized connector used to connect systems that are not Exchange servers.
- `Client <SwitchParameter>`: This parameter can be used to specify the client usage type. The usage type specifies the default permission groups and authentication methods that are assigned to this connector. The Client Receive Connector is used to receive email from users of Microsoft Exchange. It is configured to accept client submissions only from authenticated Microsoft Exchange users.
- `Internal <SwitchParameter>`: This parameter can be used to specify the internal usage type. The usage type specifies the default permission groups and authentication methods that are assigned to the Receive Connector. The Internal Receive Connector is used to receive email from servers within the Exchange organization. It is configured to accept connections only from Exchange Servers.

- ❑ **Internet** <SwitchParameter>: This parameter can be used to specify the Internet usage type. The usage type specifies the default permission groups and authentication methods that are assigned to this connector. The Internet Receive Connector is used to receive email from Internet servers. It is configured to accept connections from anonymous users.
- ❑ **Partner** <SwitchParameter>: This parameter can be used to specify the partner usage type. The usage type specifies the default permission groups and authentication methods that are assigned to this connector. The Partner Receive Connector is used to receive email from partner domains. This connector is configured to accept connections from servers that authenticate with Transport Layer Security (TLS) certificates for SMTP domains that are included in the list of domain-secured domains. You can use the `TLSReceiveDomainSecureList` parameter of the `Set-TransportConfig` cmdlet to add domains to this list.

### New-SendConnector

The `New-SendConnector` cmdlet is used to create new Send Connectors in Exchange Server 2007. Required parameters include `AddressSpaces` and `name`. The following switch parameters, `Custom`, `Partner`, `Internal`, and `Internet`, indicate the four types of Send Connectors that can be created in Exchange Server 2007. They can be specified separately or using the `Usage` parameter but cannot be used in conjunction with this parameter. If no usage type is specified, the default usage type of `custom` will be used to create the Send Connector. These switch parameters are discussed further in the following list:

- ❑ **Custom** <SwitchParameter>: This parameter can be used to specify the custom usage type. The usage type specifies no default permissions and no authentication methods to the Send Connector. This connector will be used to connect with systems that are not Exchange Servers.
- ❑ **Internal** <SwitchParameter>: This parameter can be used to specify the internal usage type. The usage type specifies the default permission groups and authentication methods that are assigned to the Send Connector. The Internal Send Connector is used to send email from servers within the Exchange organization. It is configured to route email to internal Exchange Servers as smart hosts.
- ❑ **Internet** <SwitchParameter>: This parameter can be used to specify the Internet usage type. The usage type specifies the default permissions and authentication methods that are assigned to this connector. This Send Connector is used to send emails to the Internet and configured to use DNS MX records to route messages.
- ❑ **Partner** <SwitchParameter>: This parameter can be used to specify the partner usage type. The usage type specifies the default permissions and authentication methods that are assigned to this connector. The Partner Send Connector is used to send email to partner domains. This connector is configured to only allow connections to servers that authenticate with Transport Layer Security (TLS) certificates for SMTP domains that are included in the list of domain-secured domains. You can use the `TLSReceiveDomainSecureList` parameter of the `Set-TransportConfig` cmdlet to add domains to this list.

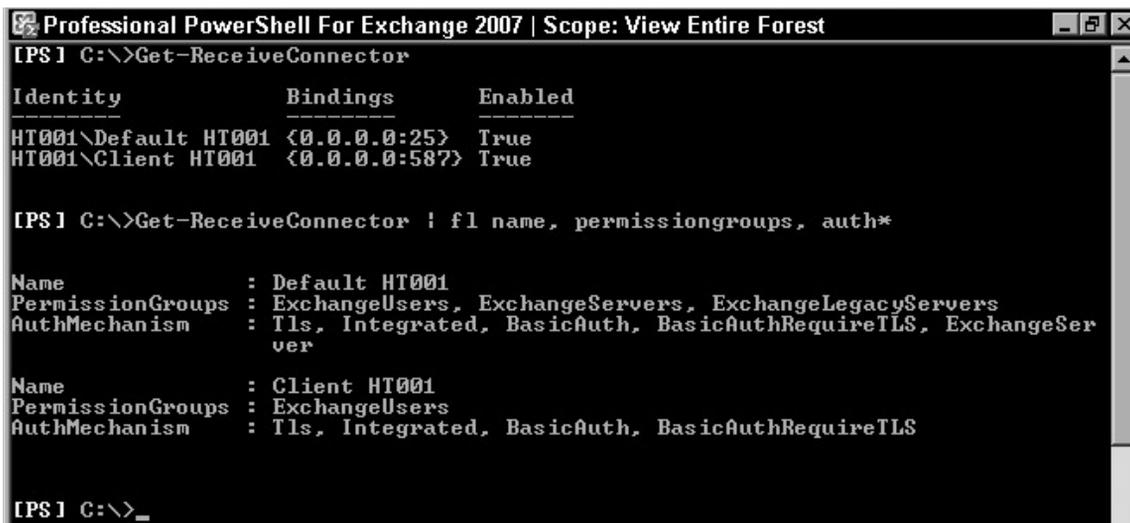
### Configuring Receive Connectors

The Receive Connector is identical in some ways to the SMTP virtual server in Exchange Server 2003. It listens for incoming SMTP connections and accepts or rejects them based on its configuration. The Receive Connector is configured with an IP address, a listening port, and the range of IP addresses that

## Part II: Working with Server Roles

can submit messages to it. In reality, though, it is the `MSExchangetransport.exe` service, otherwise known as the Process Manager, that actively listens for incoming requests. Requests are acted upon by an existing or new transport worker (`Edgetransport.exe`) process spawned by the Process Manager. `Edgetransport.exe` accepts the incoming SMTP connection, evaluates the connection criteria (IP address, port, remote IP range) and applies the session to a matching Receive Connector. Communication is always SMTP-based and each Receive Connector must have a unique combination of local IP address, port, and remote IP range configurations throughout the organization. In Exchange Server 2003, you could create multiple SMTP virtual servers but they had to be unique to either a port or IP address. In Exchange Server 2007, you can create multiple Receive Connectors with the same IP address and port; however, the remote IP range must be unique.

On a Hub Transport server, two Receive Connectors are created by default as shown in Figure 7-10. The bindings indicate that the Receive Connectors are configured to listen on all IP addresses on available network adapters on the server and on the designated ports. The Default Receive Connector is configured to listen on port 25 while the Client Receive Connector listens on port 587. The Client Receive Connector accepts SMTP connections from all non-MAPI clients, such as POP and IMAP. To view all the properties on both Receive Connectors, use the `Format-List` cmdlet in addition to the `Get-ReceiveConnector` cmdlet. A basic difference between both connectors besides the bindings is the `permissiongroups` attribute also shown in Figure 7-10. This makes sense because only non-MAPI clients connect to the Client Receive Connector.



```
[PS] C:\>Get-ReceiveConnector

Identity          Bindings          Enabled
-----
HT001\Default HT001 <0.0.0.0:25>    True
HT001\Client HT001 <0.0.0.0:587>  True

[PS] C:\>Get-ReceiveConnector | fl name, permissiongroups, auth*

Name                : Default HT001
PermissionGroups    : ExchangeUsers, ExchangeServers, ExchangeLegacyServers
AuthMechanism       : Tls, Integrated, BasicAuth, BasicAuthRequireTls, ExchangeServer

Name                : Client HT001
PermissionGroups    : ExchangeUsers
AuthMechanism       : Tls, Integrated, BasicAuth, BasicAuthRequireTls

[PS] C:\>_
```

Figure 7-10

As mentioned earlier, to create a new Receive Connector, use the `New-ReceiveConnector` cmdlet. The `Name`, `Bindings`, and `RemoteIPRanges` are required parameters. You can create a Receive Connector based on its intended use. The following switch parameters, `Custom`, `Client`, `Partner`, `Internal`, and `Internet`, indicate the types of Receive Connectors that can be created in Exchange Server 2007. They can be specified separately or using the `Usage` parameter but cannot be used in conjunction with this parameter. If no usage type is specified, the default usage type of `custom` will be set; and a Receive Connector identical in attributes to the Default Receive Connector with the exception of the specified parameters is created.

## Chapter 7: Configuring the Hub Transport Role

Note that in Exchange Server 2007 SP1, you must specify a usage type when using the `New-ReceiveConnector` cmdlet. Also note that Exchange Server 2007 SP1 supports the Internet Protocol Version 6 (IPv6) addresses and when deployed on Windows Server 2008, you can specify both IPv4 and IPv6 addresses for the `RemoteIPRange` parameter. Figure 7-11 shows how to create a new Receive Connector, and then display the connectors created with differences in authentication methods and permission groups. Some connectors have been created in advance.

```
Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>New-ReceiveConnector -Name 'RC Custom' -Bindings 172.16.8.56:25 -Remote
IPRanges 172.16.8.1-172.16.8.60 -Usage Custom

Identity          Bindings          Enabled
-----
HT001\RC Custom  <172.16.8.56:25> True

[PS] C:\>Get-ReceiveConnector

Identity          Bindings          Enabled
-----
HT001\Default HT001 <0.0.0.0:25>      True
HT001\Client HT001 <0.0.0.0:587>    True
HT001\RC Client  <172.16.8.53:587> True
HT001\RC Partner <172.16.8.53:25>  True
HT001\RC Internal <172.16.8.54:25>  True
HT001\RC Internet <172.16.8.55:25>  True
HT001\RC Custom  <172.16.8.56:25>  True

[PS] C:\>Get-ReceiveConnector | fl name, permissiongroups, auth*

Name              : Default HT001
PermissionGroups  : ExchangeUsers, ExchangeServers, ExchangeLegacyServers
AuthMechanism     : Tls, Integrated, BasicAuth, BasicAuthRequireTls, ExchangeServer

Name              : Client HT001
PermissionGroups  : ExchangeUsers
AuthMechanism     : Tls, Integrated, BasicAuth, BasicAuthRequireTls

Name              : RC Client
PermissionGroups  : ExchangeUsers
AuthMechanism     : Tls, Integrated, BasicAuth, BasicAuthRequireTls

Name              : RC Partner
PermissionGroups  : Partners
AuthMechanism     : Tls

Name              : RC Internal
PermissionGroups  : ExchangeServers, ExchangeLegacyServers
AuthMechanism     : Tls, ExchangeServer

Name              : RC Internet
PermissionGroups  : AnonymousUsers
AuthMechanism     : Tls

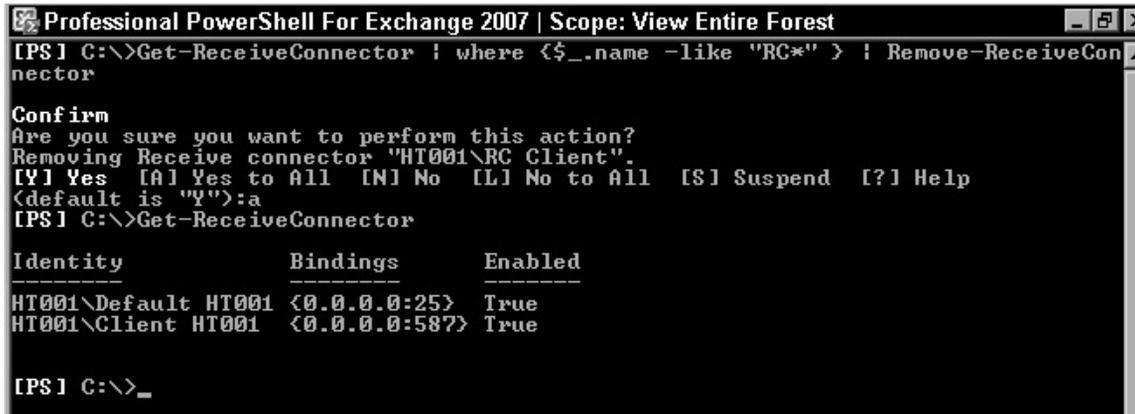
Name              : RC Custom
PermissionGroups  : None
AuthMechanism     : Tls

[PS] C:\>
```

Figure 7-11

## Part II: Working with Server Roles

Next, you use the `Remove-ReceiveConnector` cmdlet to bulk remove the connectors created, as shown in Figure 7-12.



```
Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-ReceiveConnector | where {$_.name -like "RC*" } | Remove-ReceiveConnector

Confirm
Are you sure you want to perform this action?
Removing Receive connector "HT001\RC Client".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help
<default is "Y">:a
[PS] C:\>Get-ReceiveConnector

Identity          Bindings          Enabled
-----
HT001\Default    HT001 <0.0.0.0:25> True
HT001\Client     HT001 <0.0.0.0:587> True

[PS] C:\>_
```

Figure 7-12

### Setting Relay Restrictions and Submit Permissions

After creating the Receive Connector, you can control how messages flow through the connector. In Exchange Server 2000/2003, you could configure relay restrictions to determine if messages could be relayed to users not in the Exchange organization. You could also configure permissions to determine who could submit a message to the SMTP virtual server. The same can be accomplished in Exchange Server 2007 when you specify permission groups for the connector. In Figure 7-11 you saw the permission groups assigned to each type of Receive Connector created. There are specific permissions associated with these permission groups. For example, the `ms-Exch-SMTP-Accept-Any-Recipient` permission allows the session to relay messages through the connector. To view the permissions, use the `Get-ADPermission` cmdlet in conjunction with the `Get-ReceiveConnector` cmdlet, as shown in Figure 7-13. Such granular changes can be made directly to Active Directory or by using the `Add/Remove-ADPermission` cmdlet. On the other hand, the `Set-ReceiveConnector` cmdlet can be used to add/remove permission groups.

For example, you may have created a Receive Connector with a Partner domain. By default the permission group associated with that connector allows for the partner server account to submit (`ms-Exch-SMTP-Submit`) messages and retain all the receive headers (`ms-Exch-Accept-Headers-Routing`) over a secure TLS session. If due to acquisition or some other valid reason you later choose to allow the partner server account to relay messages through the connector, then simply grant the granular permission `Ms-Exch-SMTP-Accept-Any-Recipient` or add the `ExchangeUsers` permission group to the Receive Connector, as shown in Figure 7-14.

Notice that when using the `Set-ReceiveConnector` cmdlet with permission groups, you have to write all the permissions over again and not simply the permission you want to add, or else the existing permissions are overwritten.

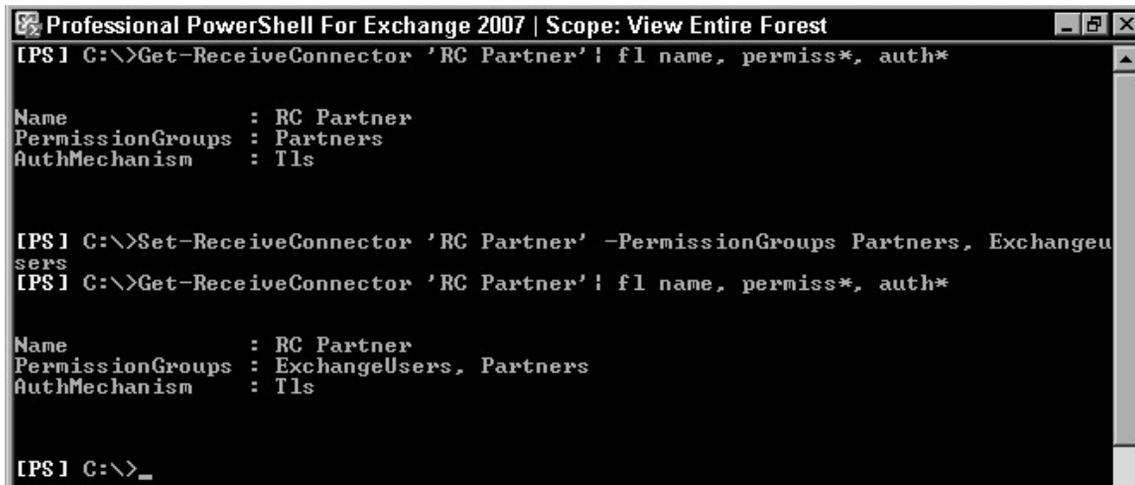
```

Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-ReceiveConnector 'HT001\Default HT001' | Get-ADPermission | ft user
, Extendedrights
User                                     ExtendedRights
-----
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-Accept-Headers-Forest>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-Accept-Headers-Organization>
NT AUTHORITY\Authenticated Users       <ms-Exch-SMTP-Submit>
NT AUTHORITY\Authenticated Users       <ms-Exch-Bypass-Anti-Spam>
NT AUTHORITY\Authenticated Users       <ms-Exch-Accept-Headers-Routing>
NT AUTHORITY\Authenticated Users       <ms-Exch-SMTP-Accept-Any-Recipient>
EXCHEXCH\Exchange Servers              <ms-Exch-SMTP-Accept-Any-Sender>
EXCHEXCH\Exchange Servers              <ms-Exch-SMTP-Accept-Exch50>
EXCHEXCH\Exchange Servers              <ms-Exch-SMTP-Accept-Authenticative-D...
EXCHEXCH\Exchange Servers              <ms-Exch-SMTP-Submit>
EXCHEXCH\Exchange Servers              <ms-Exch-Accept-Headers-Organization>
EXCHEXCH\Exchange Servers              <ms-Exch-Bypass-Message-Size-Limit>
EXCHEXCH\Exchange Servers              <ms-Exch-SMTP-Accept-Any-Recipient>
EXCHEXCH\Exchange Servers              <ms-Exch-Accept-Headers-Routing>
EXCHEXCH\Exchange Servers              <ms-Exch-Bypass-Anti-Spam>
EXCHEXCH\Exchange Servers              <ms-Exch-SMTP-Accept-Authentication-...
EXCHEXCH\Exchange Servers              <ms-Exch-Accept-Headers-Forest>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-SMTP-Accept-Any-Recipient>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-Accept-Headers-Routing>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-SMTP-Accept-Exch50>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-SMTP-Accept-Any-Sender>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-SMTP-Submit>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-SMTP-Accept-Authenticative-D...
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-Bypass-Anti-Spam>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-Bypass-Message-Size-Limit>
EXCHEXCH\ExchangeLegacyInterop         <ms-Exch-SMTP-Accept-Authentication-...
MS Exchange\Hub Transport Servers      <ms-Exch-SMTP-Accept-Authenticative-D...
MS Exchange\Hub Transport Servers      <ms-Exch-SMTP-Accept-Authentication-...
MS Exchange\Hub Transport Servers      <ms-Exch-Accept-Headers-Forest>
MS Exchange\Hub Transport Servers      <ms-Exch-Accept-Headers-Routing>
MS Exchange\Hub Transport Servers      <ms-Exch-SMTP-Accept-Any-Sender>
MS Exchange\Hub Transport Servers      <ms-Exch-Accept-Headers-Organization>
MS Exchange\Hub Transport Servers      <ms-Exch-Bypass-Anti-Spam>
MS Exchange\Hub Transport Servers      <ms-Exch-SMTP-Submit>
MS Exchange\Hub Transport Servers      <ms-Exch-SMTP-Accept-Exch50>
MS Exchange\Hub Transport Servers      <ms-Exch-SMTP-Accept-Any-Recipient>
MS Exchange\Hub Transport Servers      <ms-Exch-Bypass-Message-Size-Limit>
MS Exchange\Edge Transport Servers     <ms-Exch-SMTP-Accept-Authentication-...
MS Exchange\Edge Transport Servers     <ms-Exch-Bypass-Message-Size-Limit>
MS Exchange\Edge Transport Servers     <ms-Exch-Bypass-Anti-Spam>
MS Exchange\Edge Transport Servers     <ms-Exch-Accept-Headers-Forest>
MS Exchange\Edge Transport Servers     <ms-Exch-SMTP-Accept-Any-Recipient>
MS Exchange\Edge Transport Servers     <ms-Exch-Accept-Headers-Organization>
MS Exchange\Edge Transport Servers     <ms-Exch-SMTP-Accept-Exch50>
MS Exchange\Edge Transport Servers     <ms-Exch-Accept-Headers-Routing>
MS Exchange\Edge Transport Servers     <ms-Exch-SMTP-Accept-Any-Sender>
MS Exchange\Edge Transport Servers     <ms-Exch-SMTP-Submit>
MS Exchange\Edge Transport Servers     <ms-Exch-SMTP-Accept-Authenticative-D...

```

Figure 7-13

Several other configurable attributes exist on each Receive Connector and can be modified using the `Set-ReceiveConnector` cmdlet. These include changing the response *banner* for the server, `MaxMessageSize` limit, `MessageRateLimit`, and so forth. For a detailed list of attributes that can be configured, run `get-help set-receiveconnector` -detailed in the Exchange Management Shell.



```
Professional PowerShell For Exchange 2007 | Scope: View Entire Forest
[PS] C:\>Get-ReceiveConnector 'RC Partner' | fl name, permiss*, auth*

Name                : RC Partner
PermissionGroups    : Partners
AuthMechanism       : Tls

[PS] C:\>Set-ReceiveConnector 'RC Partner' -PermissionGroups Partners, ExchangeUsers
[PS] C:\>Get-ReceiveConnector 'RC Partner' | fl name, permiss*, auth*

Name                : RC Partner
PermissionGroups    : ExchangeUsers, Partners
AuthMechanism       : Tls

[PS] C:\>_
```

Figure 7-14

### Configuring Send Connectors

Send Connectors provide one-way outbound connections to a next hop or final destination for message delivery. They are functionally the same as SMTP connectors in Exchange Server 2000/2003 and can be configured to use DNS to route mail or forward to a smart host. They can be configured to send mail to other SMTP servers on the Internet, an Edge Transport server, or an Exchange 2000/2003 server in the same organization.

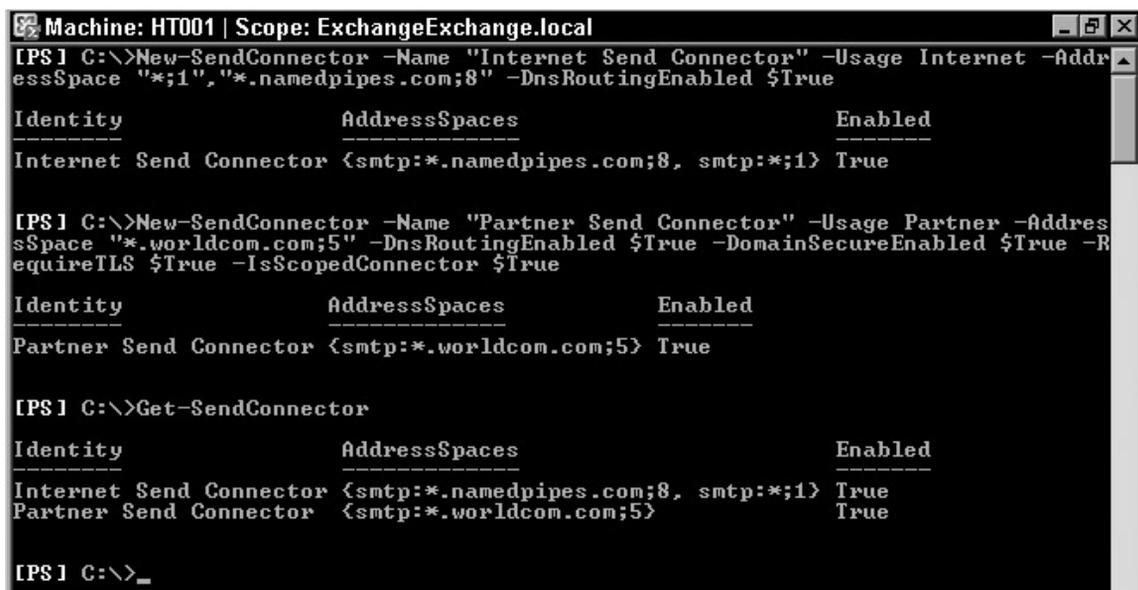
Unlike the Receive Connectors, by default when a Hub Transport or Edge Transport server is installed no explicit Send Connectors are created. However, to enable Hub Transport servers within an Exchange organization to communicate, implicit or invisible Send Connectors are automatically created. The Send Connectors are computed based on the Active Directory site topology and the site where the Hub server is installed. These implicit connectors are, however, not visible via the Exchange Management Console or Exchange Management Shell.

To create a Send Connector, use the `New-SendConnector` cmdlet. Required parameters include `AddressSpaces` and `name`. Three parts make up address spaces: the address space type, the address itself, and the cost of delivery to that address. It takes the following format: `SMTP:ExchangeExchange.local;10` representing the type, address, and cost, respectively. The Send Connector can accommodate address space types other than SMTP, such as NOTES, FAX, and so forth. Send Connectors configured with address spaces that are non-SMTP must be configured to route to a smart host and not use DNS for resolution and delivery. You can also specify multiple address spaces but they must be enclosed in double quotation marks and separated by commas.

In configuring a Send Connector, you can also specify its scope. The `IsScopedConnector` parameter set to `$True` indicates that the Send Connector is scoped to the local Active Directory site. What this means is that this connector will only be visible by Hub Transport servers that are in the same local Active Directory site as the Source servers configured on the Send Connector. It is not available to Hub servers in other Active Directory sites. The default value is `$False`. This parameter is new in Exchange Server 2007 SP1.

## Chapter 7: Configuring the Hub Transport Role

In Figure 7-15, two new connectors with different usage types are created. However, for the second connector the `IsScopedConnector` parameter limits its visibility to only Hub Transport servers in the local Active Directory site.



```
Machine: HT001 | Scope: ExchangeExchange.local
[PS] C:\>New-SendConnector -Name "Internet Send Connector" -Usage Internet -AddressSpace "*" ; 1, "*.namedpipes.com;8" -DnsRoutingEnabled $True

Identity                AddressSpaces            Enabled
-----                -
Internet Send Connector <smtp:*.namedpipes.com;8, smtp:*;1> True

[PS] C:\>New-SendConnector -Name "Partner Send Connector" -Usage Partner -AddressSpace "*.worldcom.com;5" -DnsRoutingEnabled $True -DomainSecureEnabled $True -RequireTLS $True -IsScopedConnector $True

Identity                AddressSpaces            Enabled
-----                -
Partner Send Connector  <smtp:*.worldcom.com;5> True

[PS] C:\>Get-SendConnector

Identity                AddressSpaces            Enabled
-----                -
Internet Send Connector <smtp:*.namedpipes.com;8, smtp:*;1> True
Partner Send Connector  <smtp:*.worldcom.com;5> True

[PS] C:\>_
```

Figure 7-15

Next in Figure 7-16, using the `Format-List` cmdlet provides both Send Connectors, noting the difference in scope for both connectors.



```
Machine: HT001 | Scope: ExchangeExchange.local
[PS] C:\>Get-SendConnector | fl name, max*, proto*, conne*, issc*

Name                : Internet Send Connector
MaxMessageSize      : 10MB
ProtocolLoggingLevel : None
ConnectionInactivityTimeout : 00:10:00
ConnectedDomains    : <>
IsScopedConnector   : False

Name                : Partner Send Connector
MaxMessageSize      : 10MB
ProtocolLoggingLevel : None
ConnectionInactivityTimeout : 00:10:00
ConnectedDomains    : <>
IsScopedConnector   : True

[PS] C:\>_
```

Figure 7-16

## Setting Send Connector Permissions and Authentication

Like the Receive Connector, the Send Connector is associated with certain permissions based on the type of Send Connector configured. The custom type has no default permissions. The internal connector is configured for Exchange Server Authentication and has the following permission settings: `ms-Exch-Send-Headers-Organization`, `ms-Exch-SMTP-Send-Exch50`, `ms-Exch-SMTP-Send-Exch50`, and `ms-Exch-Send-Headers-Routing`. The Internet-type Send Connector has `ms-Exch-Send-Headers-Routing` granted to the Anonymous user with no host authentication mechanism. Send Connectors connecting to Partner domains grant the Partner servers the `ms-Exch-Send-Headers-Routing` permission and can negotiate transport security.

You can modify various parameters on the Send Connector as well as remove the Send Connectors using the `Set-SendConnector` and `Remove-SendConnector` cmdlets. Figure 7-17 shows how to increase the `MaxMessageSize` of the Send Connectors, increase logging level, and modify the scope of the Partner Send Connector to be visible throughout the Exchange organization. Finally, you remove all configured connectors.



```
Machine: HT001 | Scope: ExchangeExchange.local
[PS] C:\>Get-SendConnector | Set-SendConnector -MaxMessageSize 50MB -ProtocolLog
gingLevel verbose -ConnectionInactivityTimeout 00:20:00
[PS] C:\>Set-SendConnector "Partner Send Connector" -IsScopedConnector $False
[PS] C:\>Get-SendConnector | fl name, max*, proto*, conne*, issc*

Name                : Internet Send Connector
MaxMessageSize      : 50MB
ProtocolLoggingLevel : Verbose
ConnectionInactivityTimeout : 00:20:00
ConnectedDomains    : {}
IsScopedConnector   : False

Name                : Partner Send Connector
MaxMessageSize      : 50MB
ProtocolLoggingLevel : Verbose
ConnectionInactivityTimeout : 00:20:00
ConnectedDomains    : {}
IsScopedConnector   : False

[PS] C:\>Get-SendConnector | Remove-SendConnector

Confirm
Are you sure you want to perform this action?
Removing Send connector "Internet Send Connector".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help
<default is "Y">:a
[PS] C:\>Get-SendConnector
[PS] C:\>
```

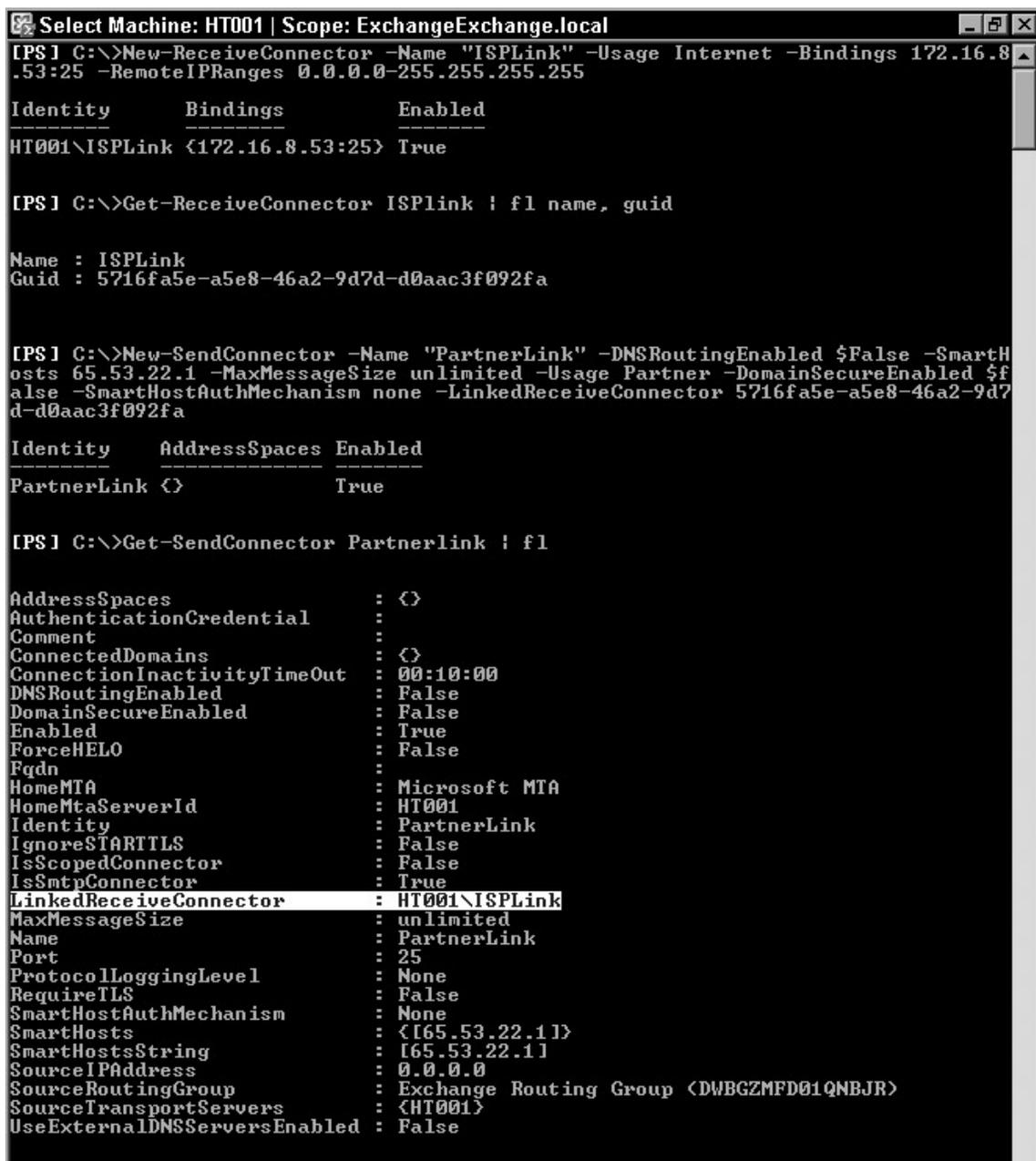
Figure 7-17

## Linking Connectors

Under certain circumstances, Send and Receive Connectors could be linked in such a way that messages received via a specific Receive Connector are always sent out via a designated Send Connector. Some organizations contract out their spam filtering solution to a vendor that checks these messages for spam or email compliance and then pushes them back into the organization for delivery. In such cases Linked Connectors could be used to direct the mail flow regardless of recipient address.

## Chapter 7: Configuring the Hub Transport Role

Figure 7-18 shows creating the Receive Connector and corresponding Send Connector, then linking both connectors. Note that you would have to specify a smart host to ensure that mail gets delivered as required.



```
Select Machine: HT001 | Scope: ExchangeExchange.local
[PS] C:\>New-ReceiveConnector -Name "ISPLink" -Usage Internet -Bindings 172.16.8.53:25 -RemoteIPRanges 0.0.0.0-255.255.255.255

Identity          Bindings          Enabled
-----
HT001\ISPLink <172.16.8.53:25> True

[PS] C:\>Get-ReceiveConnector ISPLink | fl name, guid

Name : ISPLink
Guid  : 5716fa5e-a5e8-46a2-9d7d-d0aac3f092fa

[PS] C:\>New-SendConnector -Name "PartnerLink" -DNSRoutingEnabled $False -SmartHosts 65.53.22.1 -MaxMessageSize unlimited -Usage Partner -DomainSecureEnabled $False -SmartHostAuthMechanism none -LinkedReceiveConnector 5716fa5e-a5e8-46a2-9d7d-d0aac3f092fa

Identity          AddressSpaces     Enabled
-----
PartnerLink <>          True

[PS] C:\>Get-SendConnector Partnerlink | fl

AddressSpaces           : <>
AuthenticationCredential :
Comment                 :
ConnectedDomains        : <>
ConnectionInactivityTimeout : 00:10:00
DNSRoutingEnabled       : False
DomainSecureEnabled     : False
Enabled                  : True
ForceHELO                : False
Fqdn                     :
HomeMTA                  : Microsoft MTA
HomeMtaServerId         : HT001
Identity                 : PartnerLink
IgnoreSTARTTLS           : False
IsScopedConnector       : False
IsSmtpConnector         : True
LinkedReceiveConnector   : HT001\ISPLink
MaxMessageSize           : unlimited
Name                     : PartnerLink
Port                     : 25
ProtocolLoggingLevel     : None
RequireTLS                : False
SmartHostAuthMechanism   : None
SmartHosts               : <{65.53.22.1}>
SmartHostsString         : 65.53.22.1
SourceIPAddress          : 0.0.0.0
SourceRoutingGroup       : Exchange Routing Group <DWBGZMFD01QNBJR>
SourceTransportServers   : <HT001>
UseExternalDNSServersEnabled : False
```

Figure 7-18

### Configuring a Routing Group Connector

When migrating to Exchange Server 2007 from an Existing Exchange 2000/2003 environment, there may be a period of coexistence whereby Exchange Server 2007 computers must communicate with Exchange Server 2000/2003. A Routing Group Connector is required for this communication to take place. When the first Exchange Server 2007 computer is installed into the organization, a Routing Group Connector is automatically created. To manually create this connector to another routing group, use the following command. This also enables public folder referral.

```
New-RoutingGroupConnector -Name "Coexist RGC" -SourceTransportServers  
"HT001.ExchangeExchange.local" -TargetTransportServers "ExB.ExchangeExchange.local"  
-Cost 100 -Bidirectional $true -PublicFolderReferralsEnabled $true
```

Service Pack 1 for Exchange Server 2007 introduced support for configuring a limit on the Routing Group Connector, hence the `MaxMessageSize` parameter. This parameter existed on other connectors but not the Routing Group Connector. To set the `MaxMessageSize` for an existing connector, simply pass the `Set-RoutingGroupConnector` to the output of the `Get-RoutingGroupConnector` as shown here:

```
Get-RoutingGroupConnector | Set-RoutingGroupConnector -MaxMessageSize 50MB
```

### Configuring Foreign Connectors

Foreign Connectors enable third-party vendors using non-SMTP gateways to route messages to foreign mail systems via Exchange Server 2007. This could include but is not limited to FAX systems. Exchange Server 2007 provides Foreign Connectors, which enables these messages to be placed in a drop directory to be picked up by the third-party mail system.

In the queue viewer the next hop type for these messages is identified by `NonSmtpGatewayDelivery` queue with the actual hop being the GUID of the Foreign Connector.

While the `New-ForeignConnector` cmdlet is used to create a new Foreign Connector, the `Get/Set/Remove-ForeignConnector` cmdlets can be used to view and modify the connector. Among other things that can be modified is the drop directory where messages are deposited to be picked up by the third-party's mail system. You use the following command to create a new Foreign Connector:

```
New-ForeignConnector -Name "FAX Foreign Connector" -AddressSpaces  
"FAX:*.contoso.com;5" -SourceTransportServers "HT001"
```

### Active Directory IP Site Links

Thus far, you've seen the various connectors that can be configured in Exchange Server 2007. Although connectors control the direction a message will take, Exchange Server 2007 takes advantage of the Active Directory topology and routes messages based on available Active Directory sites. In Exchange Server 2000/2003, routing was based on logical groupings of servers into routing groups. We discuss this further in Chapter 11.

Most Exchange Administrators do not have permissions to administer the Active Directory Sites and may not have control over the path that an email could traverse in a large Active Directory site. Two things can be done in this situation. An administrator may choose to specify a Hub site using the `Set-ADSite` cmdlet. This forces all mail flow in that path through the Hub Transport servers in

## Chapter 7: Configuring the Hub Transport Role

the designated Hub site before they are routed to their final destination. Something else available to an Exchange administrator is the ability to introduce an Exchange-aware attribute to existing Active Directory site links, which Exchange Server 2007 servers will consider while routing between sites. This is a cost value assigned to each IP site link evaluated by Exchange Server 2007. Messages will then be routed based on lower-cost IP site links overriding existing Active Directory site link costs. This change is made using the `Set-ADSiteLink` cmdlet.

As with the Routing Group Connector, Exchange Server 2007 SP1 now enables you to configure a `MaxMessageSize` parameter for the `ADSiteLink`. A Non Delivery Report (NDR) is generated if a message is over the specified limit.

To designate an Active Directory site as a Hub site, use the following cmdlet:

```
Set-AdSite -Identity "Site A" -HubSiteEnabled $true
```

To set the `MaxMessageSize` to the `ADSiteLink` use the following cmdlet:

```
Set-AdSiteLink -Identity DEFAULTIPSITELINK -ExchangeCost 25 -MaxMessageSize 50MB
```

## Understanding Accepted Domains and Email Address Policies

Before concluding this chapter, a few words on email address policies and accepted domains is required, because these play a huge role in email address resolution and mail routing. The following cmdlets can be used to view and change accepted domains:

- `Get-AcceptedDomain`
- `Set-AcceptedDomain`
- `New-AcceptedDomain`
- `Remove-AcceptedDomain`
- `Get-RemoteDomain`
- `Set-RemoteDomain`
- `New-RemoteDomain`
- `Remove-RemoteDomain`
- `Get-EmailAddressPolicy`
- `Set-EmailAddressPolicy`
- `New-EmailAddressPolicy`
- `Remove-EmailAddressPolicy`
- `Update-EmailAddressPolicy`

### Accepted Domains

Accepted domains are used to identify which domains the Exchange Server organization is authoritative for and thus can accept mails, and which domains it is not authoritative for and perhaps can relay mail for or return to sender. Hence, an Exchange organization will send and receive mail for an accepted domain. In Exchange Server 2007 non-authoritative domains are further segmented into internal relay domains and external relay domains. Authoritative and internal relay domains are considered to be inside the Exchange organization. Exchange accepts messages for internal relay domains and attempts to route them to a connector that can deliver the message. Both authoritative and internal relay domains can impact Email address policies and transport rules.

Exchange, on the other hand, can be configured with an external relay domain. They are identical in functionality with the internal relay domains; however, with external domains, the messages are received and processed by the Edge server, and then relayed to the destination email system. External relay domains cannot be used for email address policies.

Use the `Get-AcceptedDomain` and `Set/New/Remove-AcceptedDomain` cmdlets to view and modify the Accepted domains.

### Email Address Policies

Email address policies dictate the formatting of addresses on mail-enabled objects in the Exchange organization. Email policies formally known as Recipient Policies in Exchange 2000/2003 determine what proxy addresses are stamped on mail-enabled objects in Active Directory. Exchange Server 2007 only supports two address types: SMTP and Custom. Email address policies generate the primary and secondary email addresses for recipient objects to enable them to receive and send email.

Exchange Server 2007 has an email address policy for each user that is mail-enabled. By default, the recipient alias is used as the local part of the recipient's email address and the default accepted domain is appended after the "@" sign. Hence, a mail-enabled user will have an address such as `alias@accepteddomain.com` (`Trainee9@ExchangeExchange.local`). However, you can change how your recipients' email addresses will display. For example, you can specify that your recipients' email addresses display as `firstname.lastname@ExchangeExchange.local`.

Email addresses are applied to user and other mail-enabled objects when created. To force the email address policies to re-evaluate the mail-enabled objects, use the `Update-EmailAddressPolicy` cmdlet.

## Summary

Thus far, you have seen how much Exchange Server 2007 differs from earlier versions of Exchange in its transport functionality. The core transport architecture was re-written and transport improvements made in Exchange Server 2007 SP1 also make for reliable mail flow across an Exchange organization. The changes in core transport architecture changed the way messages are delivered in an Exchange organization. Without the presence of at least one Hub Transport server in an Active Directory site, there can be no mail flow. The Hub Transport server, as you have seen, implements the core transport functionality and serves to ensure intra-organizational email communication and compliance.

## Chapter 7: Configuring the Hub Transport Role

---

Additionally, this chapter showed the various connectors that can be created in Exchange Server 2007. Send Connectors and Receive Connectors are generally used to control message flow, however other connectors also can be configured such as Linked Connectors, Routing Group Connectors, and Foreign Connectors. With the creation of Send Connectors and Receive Connectors, several usage types are possible, which specify the authentication mechanism and permission groups to be associated with that connector. As you will recall, by default two explicit Receive Connectors are created with the installation of the Hub Transport server role and no explicit Send Connectors are installed; however, an implicit Send Connector exists to enable Hub Transport servers in the same Active Directory site exchange email messages. Hence no explicit Send Connectors are required to configure mail flow between Hub Transport servers in the same Active Directory site. Finally, we briefly reviewed the concept of accepted domains and email address policies.

