##
Lab 08 – Configuring Connectors and Transport Rules
##

Objective:
This lab focuses on managing message routing and delivery in Microsoft Exchange Server by configuring Send Connectors and Transport Rules. These tasks are essential for enabling communication with external domains, enforcing organizational policies, and customizing the flow of email messages.

You will learn to:
- Create Send Connectors using both Exchange Admin Center (EAC) and Exchange Management Shell (EMS).
- Configure different types of send connectors (e.g., Internet and Partner).
- Define address spaces and set smart host routing or DNS-based delivery.
- Verify connector functionality through mail flow testing.
- Create and test Transport Rules that apply actions such as modifying subjects or restricting delivery based on sender/recipient/domain.

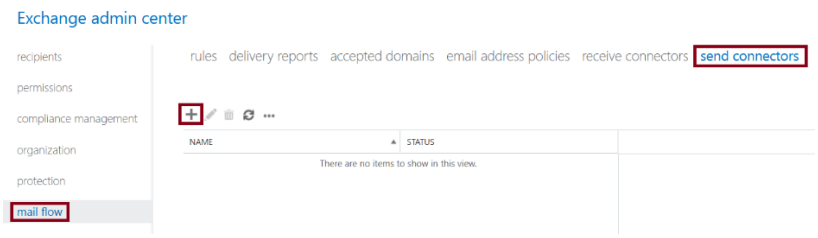
Lab Environment:
Exchange Server: ad07.domain07.local
Test External Domains: itmt.ca, external.com
Tools Used:
- Exchange Admin Center (EAC) via https://ad07.domain07.local/ecp
- Exchange Management Shell (EMS)
- Outlook or OWA for mail testing

##
Exercise 1 – Task 1 a) Create a Send Connector using the Exchange Admin Center (EAC)
##

Objective:
Create a new **Send Connector** named `ITMT Consultation` that uses **MX Registration** to route mail for the external domain `itmt.ca`. The connector type will be **Partner**, and it will use your local Exchange server as the source.

Action:
Perform the configuration from the Exchange Admin Center (EAC) interface.

Navigation:
1. Open your browser and go to: https://ad07.domain07.local/ecp
2. Log in with domain administrator credentials.
3. In the **left pane**, click on **mail flow**.
4. Click on the **Send connectors** tab.
5. Click the **+ (plus)** icon to create a new send connector.

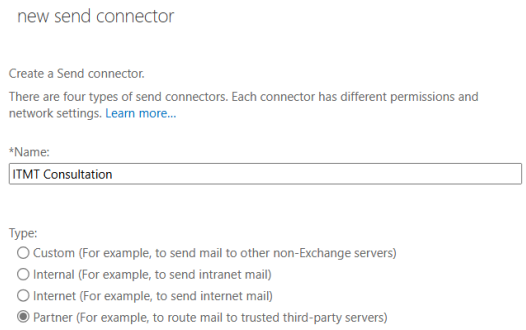


Configuration:

In the **New Send Connector** window:

- **Name**:
  Enter: `ITMT Consultation`

- **Type**:
  Select: `Partner`

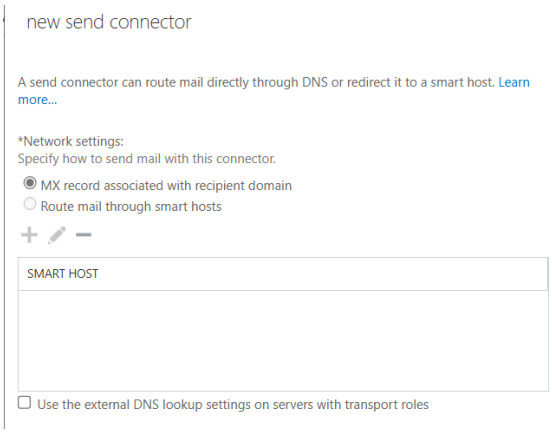


On the **Network settings** screen (shown in your screenshot):

- Choose: **MX record associated with recipient domain**
  This tells Exchange to use DNS to look up the MX record for `itmt.ca` and send mail directly there.

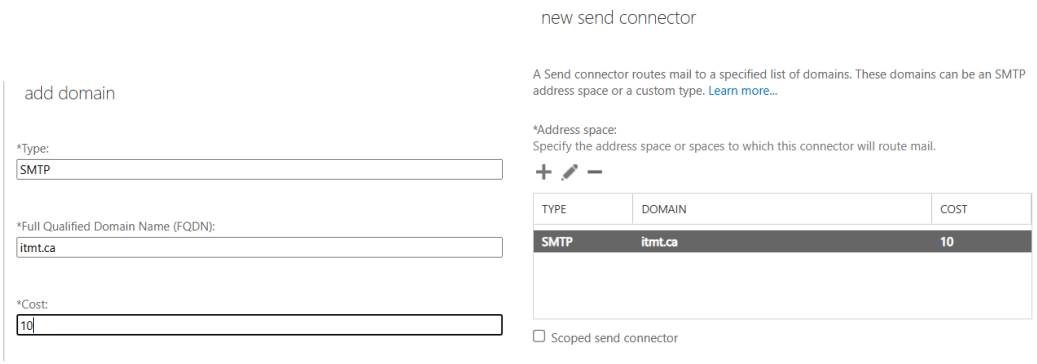
- Leave the **SMART HOST** section empty
  Do NOT add any smart host since were not routing via an intermediate server.

- Leave **Use the external DNS lookup settings on servers with transport roles** **unchecked** (unless explicitly required by your instructor).



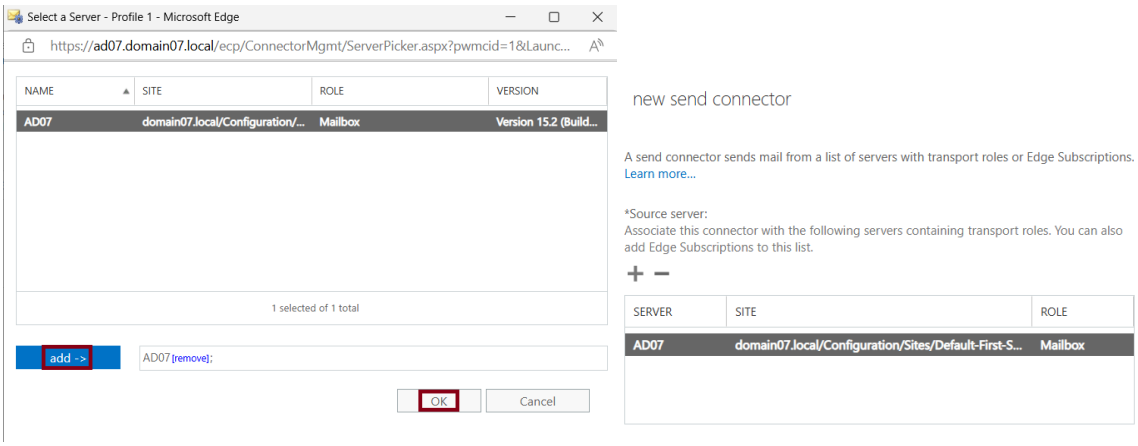
Click **Next** to **continue.**

- **Address Space**:
  Click **Add (+)**, then enter the following:
  - **Type**: `SMTP`
  - **FQDN**: `itmt.ca`
  - **Cost**: `10`



  Click **Save**.

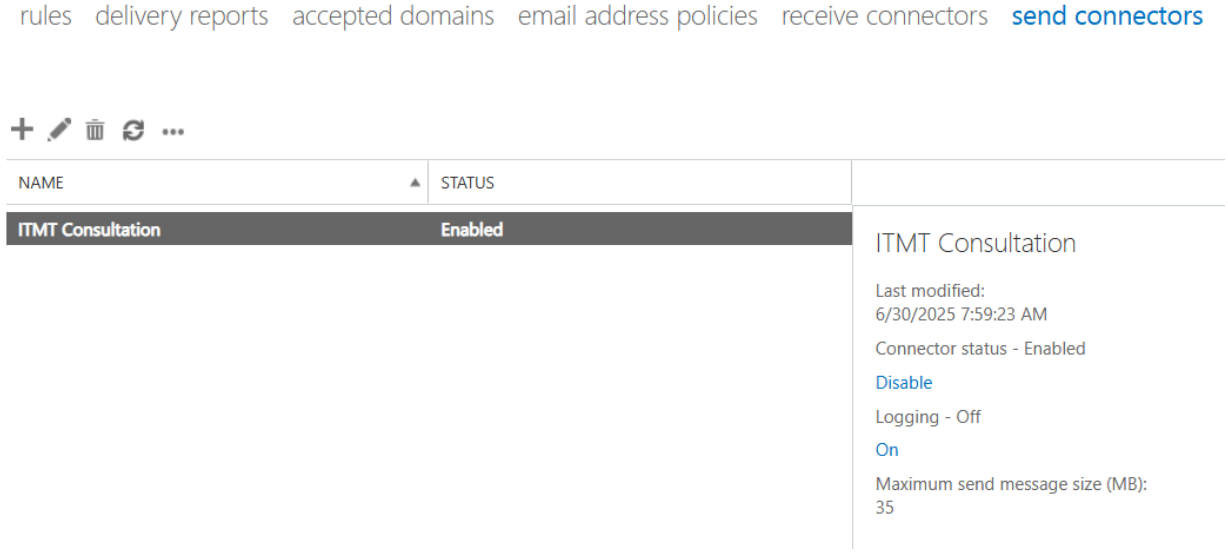
- **Source Server**:
  Click **Add**, select your Exchange Server (e.g., `ad07`), then click **Add → OK**.



6. Review the settings and click **Finish** to create the connector.

Explanation:
This send connector is used to route emails destined for the domain `itmt.ca`.
- Partner connectors are used to ensure secure delivery between organizations.
- DNS MX registration allows Exchange to resolve the mail exchanger (MX) records for `itmt.ca`.
- The address space scope ensures this connector only processes mail for the `itmt.ca` domain.
- The cost value (10) sets its delivery preference relative to other connectors.

Verification:
You should now see `ITMT Consultation` listed in the **Send connectors** tab.




```
############################################################
Exercise 1 - Task 2: Using Exchange Management Shell (EMS)
Objective: Manage Send Connectors using PowerShell
############################################################
```

Step a) List all existing Send Connectors in the organization

This command displays all configured send connectors on your Exchange server.
It helps confirm existing connectors before making changes.

```
Get-SendConnector
```

```
[PS] C:\Users\Administrator>Get-SendConnector

Identity          AddressSpaces     Enabled
--------          -------------     -------
ITMT Consultation {SMTP:itmt.ca;10} True
```

Step b) Create a new Internet Send Connector for all Address Spaces (*)

We are creating a send connector named "Internet Outbound" that allows
outbound mail flow to any domain (*).
The Usage is set to Internet, and DNSRoutingEnabled means it uses MX records
for routing.
Replace "AD07" with your actual Exchange server name if different.

```
New-SendConnector -Name "Internet Outbound" -Usage Internet `
  -AddressSpaces ("*") `
  -DNSRoutingEnabled $true `
  -SourceTransportServers "AD07"
```

```
[PS] C:\Users\Administrator>New-SendConnector -Name "Internet Outbound" -Usage Internet
>>   -AddressSpaces ("*") `
>>   -DNSRoutingEnabled $true `
>>   -SourceTransportServers "AD07"

Identity          AddressSpaces Enabled
--------          ------------- -------
Internet Outbound {smtp:*;1}    True
```

Note:
There is also a simplified version of the command:
New-SendConnector -Internet -Name Internet -AddressSpaces *

Both commands achieve a similar result, but the full version gives you more control and visibility into:
- The connector's usage type
- DNS-based routing behavior
- The specific source transport server
This makes it easier to understand and maintain in larger environments.

Step c) List all send connectors to verify the new Internet connector

This command displays all existing send connectors in your Exchange organization,including their name, address space, usage, and status. Use it to confirm that "Internet Outbound" was successfully created in the previous step.

```
Get-SendConnector | Format-Table Name,AddressSpaces,Usage,Enabled
```

```
[PS] C:\Users\Administrator>Get-SendConnector | Format-Table Name,AddressSpaces,Usage,Enabled

Name              AddressSpaces     Usage Enabled
----              -------------     ----- -------
ITMT Consultation {SMTP:itmt.ca;10}         True
Internet Outbound {smtp:*;1}                True
```

Explanation:
- Get-SendConnector: Retrieves all defined send connectors.
- Format-Table: Displays selected properties in a readable table.
- Name: Shows the connector name.
- AddressSpaces: Shows which domain(s) the connector applies to.
- Usage: Indicates whether the connector is for internal, partner, or internet mail.
- Enabled: Displays if the connector is active.

Note:
This command is ideal for visual validation. For scripting or automation scenarios,
you can use the following version for more structured output:

```
Get-SendConnector | Select-Object Identity,Enabled
```

```
[PS] C:\Users\Administrator>Get-SendConnector | Select-Object Identity, Enabled

Identity          Enabled
--------          -------
ITMT Consultation    True
Internet Outbound    True                                                    A
```

Step d) Disable the ITMT Consultation send connector

This step disables the send connector named "ITMT Consultation" to temporarily prevent it
from routing messages **while** keeping the configuration intact.

```
Set-SendConnector -Identity "ITMT Consultation" -Enabled $false
```

```
[PS] C:\Users\Administrator>Set-SendConnector -Identity "ITMT Consultation" -Enabled $false
```

Explanation:
- Set-SendConnector: Modifies properties of an existing send connector.
- -Identity: Specifies the name of the connector you want to modify.
- -Enabled **$false**: Disables the connector without deleting it, so it can be re-enabled later **if** needed.

Tip:
Disabling a connector is useful during testing, troubleshooting, or when you want to
redirect traffic through a different route without losing your original configuration.

Step e) Check that the ITMT Consultation send connector has been disabled

This step confirms whether the connector is currently active or not.

```
Get-SendConnector -Identity "ITMT Consultation" | Format-List Name,Enabled
```

```
[PS] C:\Users\Administrator>Get-SendConnector -Identity "ITMT Consultation" | Format-List Name,Enab
led


Name    : ITMT Consultation
Enabled : False
```

Explanation:
- Get-SendConnector: Retrieves the specified send connector.
- -Identity: Filters the result to only show "ITMT Consultation".
- Format-List: Displays the selected properties **in** a vertical format **for** clarity.
- Name: Displays the connector's name.
- Enabled: Should **return** **False** **if** the connector is correctly disabled.

Note:
If the result shows `Enabled : False`, the connector is successfully disabled. **If** it shows `True`, the disable command from Step d may not have applied correctly.

Step f) Delete the ITMT Consultation send connector

We are now removing the "ITMT Consultation" connector from the Exchange configuration.

```
Remove-SendConnector -Identity "ITMT Consultation" -Confirm:$false
```

```
[PS] C:\Users\Administrator>Remove-SendConnector -Identity "ITMT Consultation" -Confirm:$false
[PS] C:\Users\Administrator>
```

Explanation:
- Remove-SendConnector: Deletes the specified send connector from your Exchange organization.
- -Identity: Specifies the connector to delete, here "ITMT Consultation".
- -Confirm:$false: Skips the interactive confirmation prompt to streamline automation or scripting.

Important:
Use `-Confirm:$false` carefully. Without it, PowerShell would ask for confirmation before deleting. This flag forces deletion immediately, which is useful in scripted environments but should be used with caution to avoid accidental removal.

Step g) Verify that the ITMT Consultation send connector has been removed

After deletion, we confirm that the connector named "ITMT Consultation" no longer exists in the Exchange configuration.

```
Get-SendConnector | Where-Object { $_.Name -eq "ITMT Consultation" }
```

```
[PS] C:\Users\Administrator>Get-SendConnector | Where-Object { $_.Name -eq "ITMT Consultation" }
[PS] C:\Users\Administrator>Get-SendConnector

Identity          AddressSpaces Enabled
--------          ------------- -------
Internet Outbound {smtp:*;1}    True
```

Explanation:
- Get-SendConnector: Retrieves all configured send connectors.
- Where-Object { $_.Name -eq "ITMT Consultation" }: Filters the result to check if a connector with the specified name still exists.
- If no output appears, it confirms that the connector was successfully deleted.

Alternative (for a quick Boolean result):
(Get-SendConnector | Where-Object { $_.Name -eq "ITMT Consultation" }) -eq $null

```
[PS] C:\Users\Administrator>(Get-SendConnector | Where-Object { $_.Name -eq "ITMT Consultation" })
-eq $null
True
```

→ This returns `True` if the connector is fully removed.

##
Exercise 2 – Configuring Receive Connectors
##

Objective:
In this exercise, we will configure a custom receive connector in Exchange to allow SMTP connections from a trusted application (e.g., SharePoint) using a non-default port (2525). This setup simulates how third-party services securely send mail through the Exchange server.

Explanation:
Receive connectors are used to accept incoming SMTP messages into the Exchange transport pipeline. They control how and from where mail can be received. By default, Exchange has standard receive connectors for internal and external mail flow.

However, when integrating third-party systems such as SharePoint, printers, or backup systems, you often need a dedicated receive connector with custom settings:
- Binding it to a specific IP and port (e.g., 192.168.X.1:2525)
- Allowing authentication using specific methods (e.g., Basic Authentication)
- Restricting or allowing IP ranges (e.g., allowing all or only certain devices)

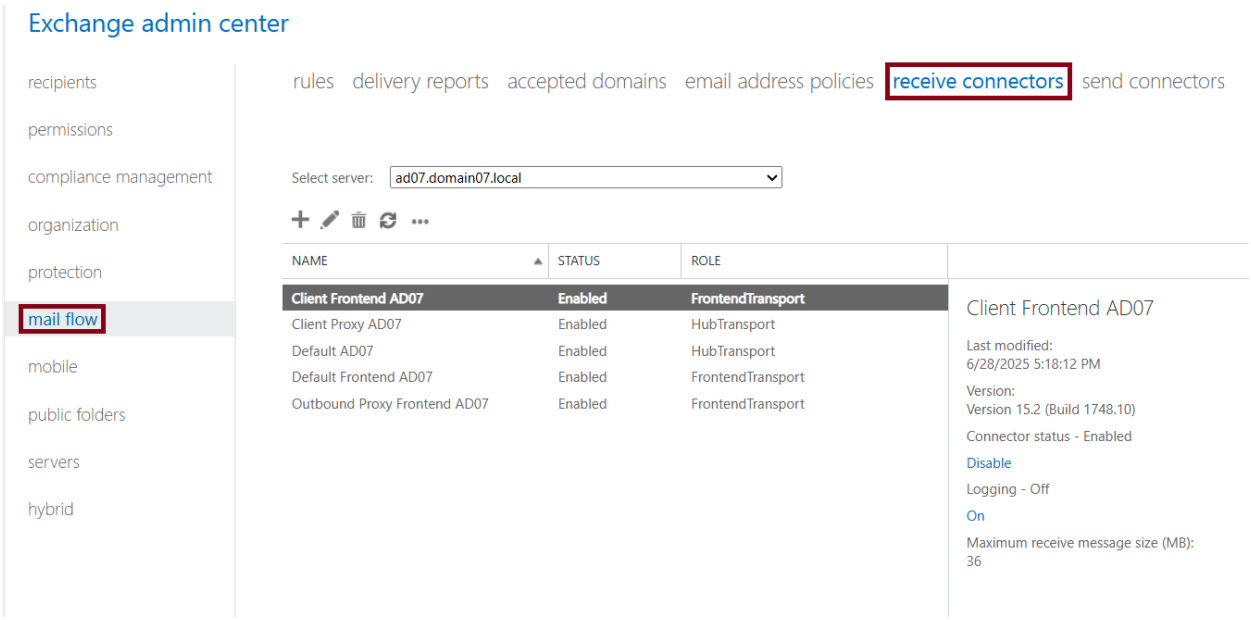
This exercise will guide you through setting up a receive connector named "SharePoint" that accepts mail on port 2525 using Basic Authentication, and permits messages from any IP address, simulating a flexible yet controlled integration point.

##
Step 1a – Check the Existing Receive Connectors
##

Action:
View the current receive connectors configured on the Exchange server.

Navigation:
In the Exchange Admin Center (EAC), go to:
Mail Flow → Receive Connectors

Explanation:
This section displays all receive connectors currently defined for the selected server (e.g., ad07.domain07.local).
These connectors control how incoming SMTP messages are accepted by the server.



By default, Exchange Server automatically creates several built-in receive connectors during installation:
- **Client Frontend AD07**: Used by authenticated SMTP clients (like Outlook) to send mail via port 587.
- **Client Proxy AD07**: Used by internal Exchange servers to relay messages on behalf of clients.
- **Default AD07**: Used for mail flow between Exchange servers.
- **Default Frontend AD07**: Listens on port 25 and accepts anonymous connections from external senders.
- **Outbound Proxy Frontend AD07**: Optional connector used when proxying outbound mail through a frontend server.

These default connectors are essential for normal mail flow and should not be modified unless specifically required.

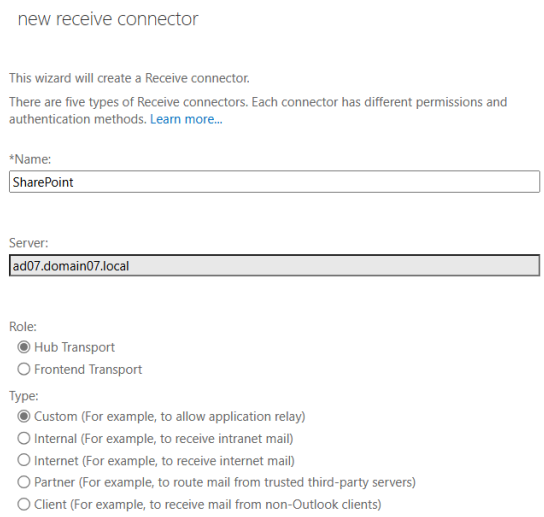
In the next steps, we will create a new **custom** connector named **SharePoint** to simulate secure communication from a trusted app to the Exchange server on a non-standard port (2525).

##
Step 1b – Create a New Receive Connector for SharePoint
##

Action:
Create a custom receive connector named "SharePoint" with specific network and binding settings.

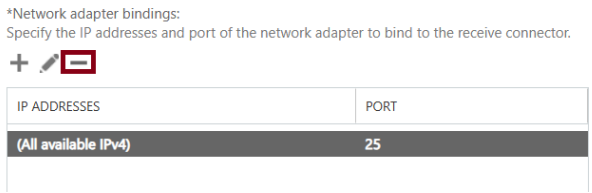
Navigation:
In the Exchange Admin Center (EAC), go to:
Mail Flow → Receive Connectors → Click the **(+)** icon to create a new connector.

Configuration Settings:
- **Name**: SharePoint
- **Role**: Hub Transport
- **Type**: Custom

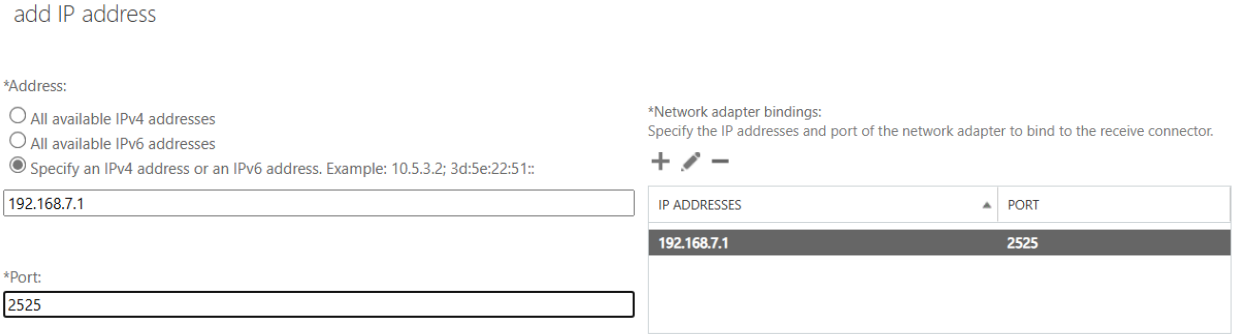


Explanation:
We are creating a dedicated receive connector to simulate SMTP connections from a SharePoint server
or similar application. By choosing the **Custom** type, we can manually define the network bindings
and allowed remote IPs **for** tighter control.

Network Adapter Bindings:
- Remove the default "All Available IPv4" entry.

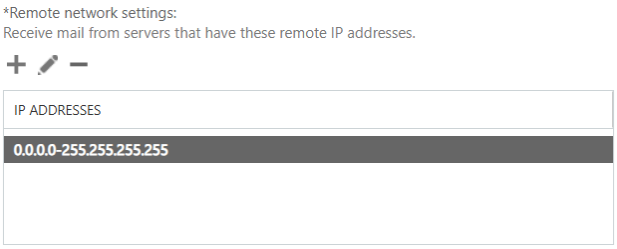


- Click **Add (+)** and enter the following:
  - **IP Address**: `192.168.7.1`
  - **Port**: `2525`



Explanation:
This means the connector will listen for SMTP traffic on port 2525, but only from the NIC with IP address `192.168.X.1`.
Using a non-standard port (like 2525) is common when integrating third-party apps to avoid conflicts
with the default SMTP listener on port 25.

Remote Network Settings:
- Leave this to default: **All IP addresses (0.0.0.0 - 255.255.255.255)**



Explanation:
This allows any system to connect to this connector. In a real-world setup, you would restrict this to
trusted app IPs (like SharePoint or scanners), but for lab/testing purposes, the default is acceptable.

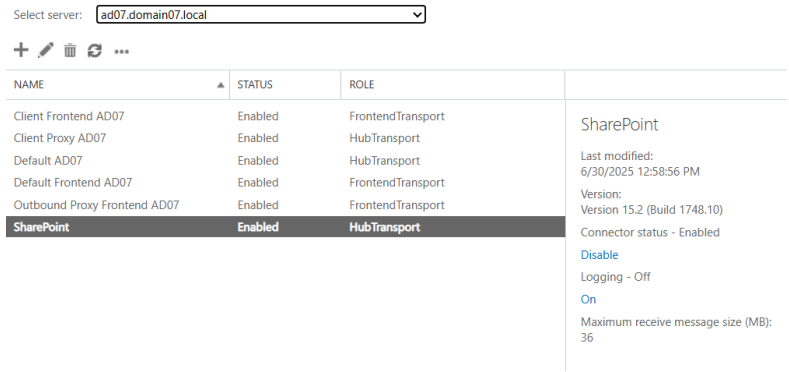
Once all settings are applied, click **Save** to create the new connector.

##
Step 1c - Save the New Receive Connector
##

Action:
Finalize and save the new SharePoint receive connector with the configured settings.

Navigation:
After configuring the following in the New Receive Connector window:
- Name: SharePoint
- Role: Hub Transport
- Type: Custom
- Network Adapter Binding: 192.168.X.1 on port 2525
- Remote Network Settings: All IP Addresses

Click the **Save** button at the bottom of the configuration window.

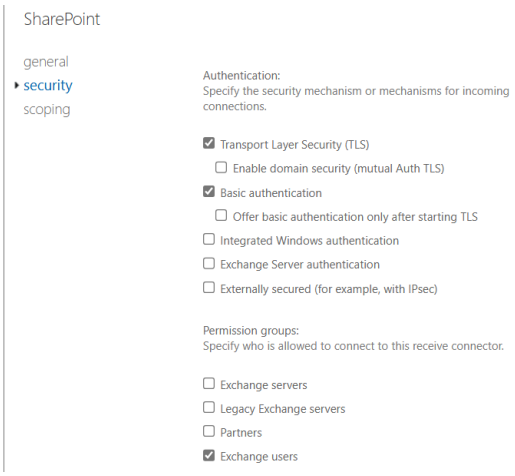


Explanation:
Saving the connector applies the configuration and makes it active immediately. The receive connector
is now ready to accept SMTP traffic on port 2525 from the local NIC IP specified. However, we still
need to configure authentication and permission groups in the next step to allow secure message
reception from trusted sources like SharePoint or other applications.

##
Step 1d – Configure Authentication and Permission Groups for SharePoint Connector
##

Action:
Modify the new SharePoint receive connector to allow basic authentication and define the appropriate permission group.

Navigation:
1. Go to **mail flow** > **receive connectors**.
2. Select the **SharePoint** connector you just created and click the **Edit (✎)** icon.
3. In the **Security** tab under Authentication :
   - Check the box for **Basic authentication**.
4. In the **Permission Groups** tab:
   - Check the box for **Exchange users**.
5. Click **Save**.



Explanation:
Enabling **Basic authentication** allows applications like SharePoint to submit mail using clear-text credentials (usually over a secure channel like TLS).
The **Exchange users** permission group grants authenticated users or services the ability to relay messages internally, which is required for integrated application scenarios.

Caution:
Basic authentication should only be enabled if youre certain the communication will occur over a secured connection (e.g., TLS), as it exposes credentials otherwise.

##
Exercise 2 – Part 2: Configuring Receive Connectors via EMS
##

Objective:
In this section, we configure and manage Receive Connectors directly using the Exchange Management Shell (EMS). This approach allows administrators to efficiently create, modify, and delete connectors using automation-friendly commands. We will create a custom Internet connector, apply anonymous permissions, and manage existing connectors via PowerShell.

##
Step a) List the existing receiving connectors in your Exchange organization
##

Action:
Display all currently configured receive connectors with their key settings.

Command:
```
Get-ReceiveConnector | Format-Table Name, Bindings, RemoteIPRanges
```

```
[PS] C:\Users\Administrator>Get-ReceiveConnector | Format-Table Name, Bindings, RemoteIPRanges

Name                           Bindings                    RemoteIPRanges
----                           --------                    --------------
Default AD07                   {0.0.0.0:2525, [::]:2525} {::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:f...
Client Proxy AD07              {[::]:465, 0.0.0.0:465}   {::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:f...
Default Frontend AD07          {[::]:25, 0.0.0.0:25}     {::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:f...
Outbound Proxy Frontend AD07 {[::]:717, 0.0.0.0:717}   {::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:f...
Client Frontend AD07           {[::]:587, 0.0.0.0:587}   {::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:f...
SharePoint                     {192.168.7.1:2525}        {0.0.0.0-255.255.255.255}
```

Explanation:
This command lists all existing receive connectors and displays their name, IP bindings, allowed remote IP ranges, and whether the connector is currently enabled. It provides a quick and complete overview that is both informative and script-friendly.

- Get-ReceiveConnector
  Retrieves all receive connectors defined in the Exchange organization.

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

Identity                          Bindings                    Enabled
--------                          --------                    -------
AD07\Default AD07                 {0.0.0.0:2525, [::]:2525} True
AD07\Client Proxy AD07            {[::]:465, 0.0.0.0:465}   True
AD07\Default Frontend AD07        {[::]:25, 0.0.0.0:25}     True
AD07\Outbound Proxy Frontend AD07 {[::]:717, 0.0.0.0:717}   True
AD07\Client Frontend AD07         {[::]:587, 0.0.0.0:587}   True
AD07\SharePoint                   {192.168.7.1:2525}        True
```

• Format-Table Name, Bindings, RemoteIPRanges, Enabled
  Displays the output in a structured table showing:
  - Name: The connector name.
  - Bindings: The IP/port combinations the connector listens on.
  - RemoteIPRanges: The IP ranges allowed to send mail through this connector.

Why this format:
Using just `Get-ReceiveConnector` by itself returns a verbose object with too much nested data for quick review. Adding `Format-Table` with selected properties gives a cleaner view that's easier to scan, document, or validate — especially useful when dealing with many connectors or writing scripts.

Optional Note for scripting:
This format is also ideal when piping into logs or automation reports, allowing you to include only the most relevant fields.

##
Step b) Create a new receive connector named "Internet"
##

Objective:
Create a new receive connector called "Internet" for external mail flow.
It will listen on IP 192.168.7.0 and port 2525, accepting mail from any IP.

Action:
Use the New-ReceiveConnector cmdlet with -Usage set to Internet,
and configure the Binding and RemoteIPRanges as specified.

Command:
```
New-ReceiveConnector -Name "Internet" -Server AD07 `
  -Bindings "192.168.7.0:2525" `
  -RemoteIPRanges "0.0.0.0-255.255.255.255" `
  -Usage Internet
```

```
[PS] C:\Users\Administrator>New-ReceiveConnector -Name "Internet" -Server AD07 `
>>   -Bindings "192.168.7.0:2525" `
>>   -RemoteIPRanges "0.0.0.0-255.255.255.255" `
>>   -Usage Internet

Identity      Bindings           Enabled
--------      --------           -------
AD07\Internet {192.168.7.0:2525} True
```

Explanation:
The connector "Internet" is hosted on server AD07.
It listens on IP 192.168.7.0, port 2525 for incoming SMTP connections.
It allows connections from all remote IPs (the full IPv4 range).
The usage type is set to Internet, which is ideal for third-party systems or external SMTP sources that need to relay messages through Exchange.

Optional Note:
This command is well-suited for scripting or automation in large-scale environments.

##
Step c) Enable Anonymous Users for the Internet Receive Connector
##

Objective:
Allow anonymous email systems (like external mail servers) to submit messages through the "Internet" receive connector by enabling the AnonymousUsers permission group.

Action:
Assign the AnonymousUsers group to the "Internet" connector using the Set-ReceiveConnector cmdlet.

Command:
```
Set-ReceiveConnector -Identity "AD07\Internet" -PermissionGroups AnonymousUsers
```

```
[PS] C:\Users\Administrator>Set-ReceiveConnector -Identity "AD07\Internet" -PermissionGroups Anonym
ousUsers
[PS] C:\Users\Administrator>_
```

Explanation:
- Set-ReceiveConnector: Modifies configuration settings on a specific connector.
- -Identity "AD07\Internet": Specifies the target receive connector by its full name.
- -PermissionGroups AnonymousUsers: Enables the built-in group that allows unauthenticated SMTP clients to submit mail.

Note:
An alternative way to allow anonymous relay is to assign specific extended rights using the `Add-ADPermission` cmdlet:

```
Get-ReceiveConnector "AD07\Internet" | Add-ADPermission `
  -User "NT AUTHORITY\ANONYMOUS LOGON" `
  -ExtendedRights "Ms-Exch-SMTP-Accept-Any-Recipient"
```

This method is typically used in **advanced scenarios** where:
- You want fine-grained control over SMTP permissions.
- You are configuring **relay permissions** beyond basic anonymous submission.
- You are working in environments where **custom security settings** are required.

However, for basic anonymous mail acceptance, `Set-ReceiveConnector -PermissionGroups AnonymousUsers` is easier, more transparent, and preferred in most environments.

##
Step d) Verify Anonymous Access is Enabled on the Internet Receive Connector
##

Objective:
Confirm that the "Internet" receive connector now allows anonymous SMTP connections
by checking the assigned permission groups.

Action:
Use the `Get-ReceiveConnector` cmdlet to inspect the `PermissionGroups` setting
of the "Internet" connector.

Command:
```
(Get-ReceiveConnector -Identity "AD07\Internet").PermissionGroups
```

```
[PS] C:\Users\Administrator>(Get-ReceiveConnector -Identity "AD07\Internet").PermissionGroups
AnonymousUsers
[PS] C:\Users\Administrator>
```

Explanation:
- Get-ReceiveConnector: Retrieves the configuration of the specified connector.
- -Identity "AD07\Internet": Specifies the connector to inspect.
- .PermissionGroups: Displays the permission groups assigned to the connector.
  If **AnonymousUsers** is listed, the connector allows mail from anonymous sources.

Expected Output:
AnonymousUsers

##
Step e) Delete the SharePoint Receive Connector
##

Objective:
Remove the unused or unnecessary “SharePoint” receive connector to streamline connector management and reduce potential security exposure.

Action:
Use the `Remove-ReceiveConnector` cmdlet to delete the connector named "SharePoint".

Command:
```
Remove-ReceiveConnector -Identity "AD07\SharePoint"
```

```
[PS] C:\Users\Administrator>Remove-ReceiveConnector -Identity "AD07\SharePoint"

Confirm
Are you sure you want to perform this action?
Removing Receive connector "AD07\SharePoint".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [?] Help (default is "Y"): A
[PS] C:\Users\Administrator>
```

Explanation:
- Remove-ReceiveConnector: Deletes the specified receive connector from the server.
- -Identity "AD07\SharePoint": Refers to the full identity of the connector to be removed.
  The format "ServerName\ConnectorName" ensures precision when targeting the object.

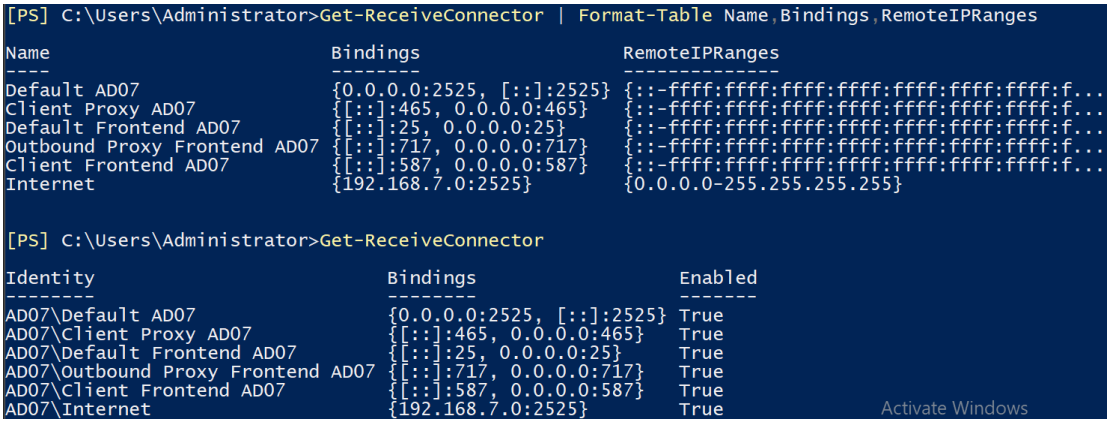
Confirmation:
You may be prompted to confirm the deletion. Press **Y** to proceed when asked.

##
Step f) Verify that the SharePoint Receive Connector Has Been Deleted
##

Objective:
Confirm that the "SharePoint" receive connector has been successfully removed from the server.

Action:
List all receive connectors and check that "SharePoint" is no longer present.

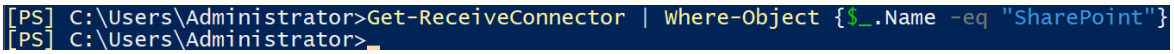
Command:
```
Get-ReceiveConnector | Format-Table Name,Bindings,RemoteIPRanges
```



Explanation:
- Get-ReceiveConnector: Retrieves all existing receive connectors in the Exchange organization.
- Format-Table: Formats the output in a readable table view.
- Name: Displays the connector name.
- Bindings: Shows the IP and port bound to the connector.
- RemoteIPRanges: Indicates the IP ranges permitted to send through the connector.

Optional Check:
If needed, you can specifically search for the "SharePoint" connector to confirm its absence:

```
Get-ReceiveConnector | Where-Object {$_.Name -eq "SharePoint"}
```



##
Exercise 3 – Configuring Transport Rules
Task 1 – Create a Transport Rule Using the Exchange Admin Center (EAC)
##

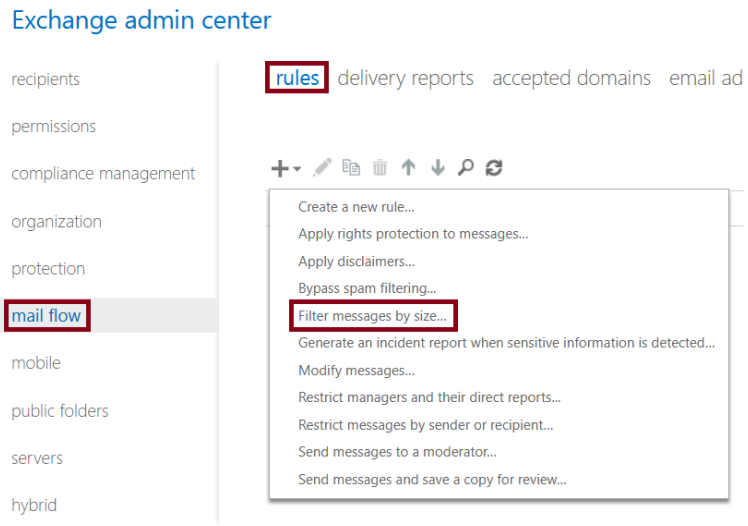
Objective:
Prevent the delivery of email messages larger than 20 MB by defining a transport rule in Exchange Admin Center (EAC). This helps manage system performance and avoid disruptions from oversized attachments.

Explanation:
Exchange transport (mail flow) rules allow administrators to apply conditions to messages in transit. In this task, we will configure a rule that blocks messages equal to or larger than 20,000 KB and informs the sender why it was blocked.

Step a) Access Transport Rules via the Menu

Action:
Begin creating a rule from a predefined template.

Navigation:
Exchange Admin Center (EAC) → **Mail Flow** → **Rules**
Click the **(+) Add** dropdown → Select **Filter** messages by size…**

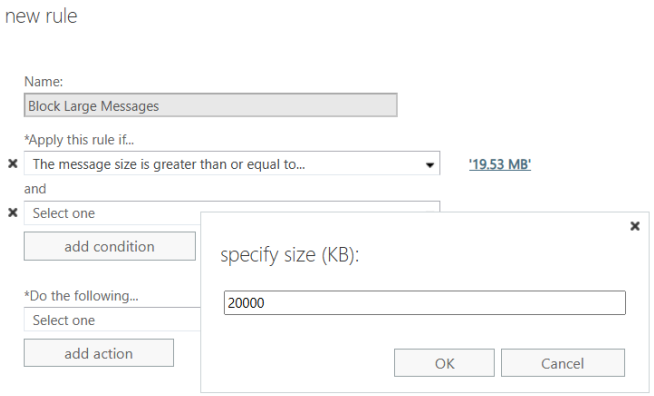


Explanation:
Instead of choosing "Create a new rule…", selecting **"Filter messages by size…"** pre-fills the condition template needed to **filter** emails by size. This simplifies the creation **process for** rules targeting message limits.

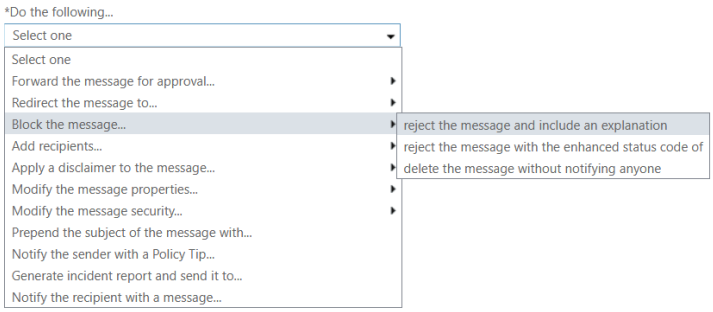
Step b) Complete Rule Configuration

Fill out the fields as follows:

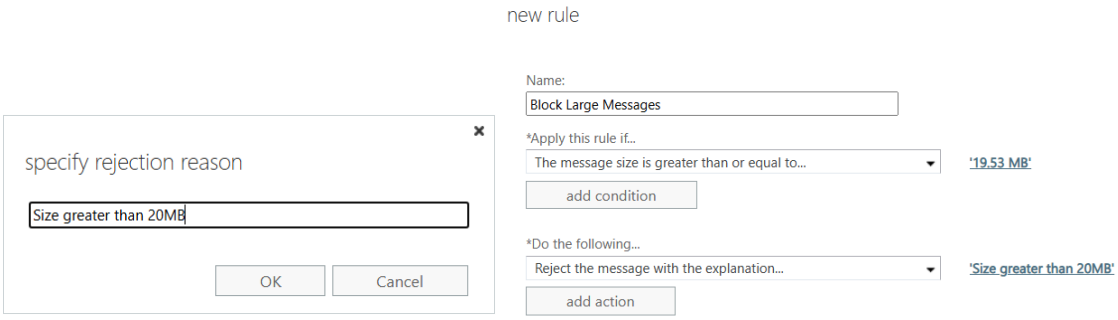
- **Name:** Block Large Messages
- **Apply this rule if:** The message size is greater than or equal to **20000 KB**



- **Do the following:** Block the message → **Reject the message and include an explanation**

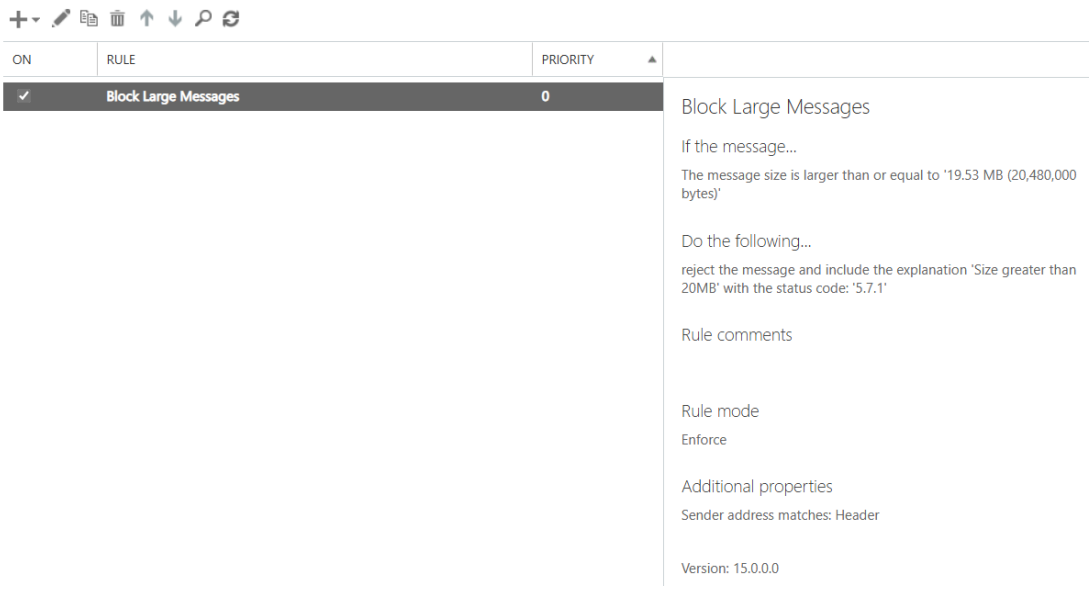


- **Explanation text:** Size greater than 20MB



- **Click Save**

Explanation:
The rule blocks any incoming or outgoing messages ≥20MB and displays a friendly rejection notice. This keeps transport queues clean and informs users of the policy.



Validation:
Once saved, the rule named **Block Large Messages** should appear in the Rules list. You can sort or filter the rules list to quickly locate and confirm its settings.

```
################################################################################
Exercise 3 - Configuring Transport Rules
Task 2 - Using Exchange Management Shell (EMS)
################################################################################
```

Objective:
Use EMS to manage transport rules including listing, creating, and removing rules. This exercise simulates typical tasks an Exchange administrator would perform to maintain compliance, manage mail flow policies, and implement legal disclaimers.

```
################################################################################
Step a) List Existing Transport Rules
################################################################################
```

Action:
Display all current transport rules configured in the Exchange organization.

Command:
```
Get-TransportRule
```

```
[PS] C:\Users\Administrator>Get-TransportRule

Name                State   Mode    Priority Comments
----                -----   ----    -------- --------
Block Large Messages Enabled Enforce 0        ...
```

Explanation:
- Get-TransportRule: Retrieves a list of all transport rules in the organization.
  This helps in auditing configurations, verifying if a rule already exists, or reviewing policy priorities.

```
################################################################################
Step b) Create a Transport Rule with a Disclaimer
################################################################################
```

Purpose:
Ensure that all outgoing emails sent outside the organization include a legal or informational disclaimer. This helps meet compliance, legal, or branding requirements by automatically appending a consistent message.

Action:
Add a new transport rule that appends a legal disclaimer to all outgoing email messages.

Used Command:
```
New-TransportRule -Name "Add Legal Disclaimer" `
  -SentToScope NotInOrganization `
  -ApplyHtmlDisclaimerText "<p><b>NOTICE:</b> This message may contain
confidential information.</p>" `
  -ApplyHtmlDisclaimerFallbackAction Wrap `
  -ApplyHtmlDisclaimerLocation Append
```

```
[PS] C:\Users\Administrator>New-TransportRule -Name "Add Legal Disclaimer" `
>>   -SentToScope NotInOrganization `
>>   -ApplyHtmlDisclaimerText "<p><b>NOTICE:</b> This message may contain confidential information.
</p>" `
>>   -ApplyHtmlDisclaimerFallbackAction Wrap `
>>   -ApplyHtmlDisclaimerLocation Append

Name                State   Mode    Priority Comments
----                -----   ----    -------- --------
Add Legal Disclaimer Enabled Enforce 1
```

Explanation:
- `-Name "Add Legal Disclaimer"`: Gives the rule a clear, descriptive name for easy management.
- `-SentToScope NotInOrganization`: Applies the rule only to external (non-org) recipients.
- `-ApplyHtmlDisclaimerText`: Specifies the HTML text of the disclaimer to append.
- `-ApplyHtmlDisclaimerFallbackAction Wrap`: Ensures email delivery by wrapping the original message if the disclaimer cannot be appended directly.
- `-ApplyHtmlDisclaimerLocation Append`: Appends the disclaimer to the bottom of the email.

Note:
This version uses a professional disclaimer message and naming convention suitable for production environments or portfolio examples. The command omits `-Enabled $true` because its implied by default when creating a new rule.

Alternate Command (Used in Course Slides):

```
New-TransportRule -Name Disclaimer `
  -Enabled $true `
  -SentToScope 'NotInOrganization' `
  -ApplyHtmlDisclaimerLocation 'Append' `
  -ApplyHtmlDisclaimerText "<h3>Disclaimer</h3><p>Cela est un
disclaimer.</p>" `
  -ApplyHtmlDisclaimerFallbackAction Wrap
```

Explanation:
- `-Name Disclaimer`: A simpler rule name used for demonstration purposes.
- `-Enabled $true`: Explicitly enables the rule; redundant but harmless.
- `-ApplyHtmlDisclaimerText`: Uses placeholder disclaimer content in French.
- Other parameters function the same as above.

Comparison:
While both commands are technically valid, the version used in this lab adopts clearer naming and a realistic disclaimer message. This improves clarity, professionalism, and portfolio value while still demonstrating command mastery.

##
Step c) List Transport Rules to Verify Disclaimer Rule Creation
##

Objective:
Confirm that the disclaimer transport rule was successfully created and is currently active.

Action:
List all existing transport rules and review their names, states, and priorities.

Command:
Get-TransportRule

```
[PS] C:\Users\Administrator>Get-TransportRule

Name                 State   Mode    Priority Comments
----                 -----   ----    -------- --------
Block Large Messages Enabled Enforce 0        ...
Add Legal Disclaimer Enabled Enforce 1
```

Explanation:
- This cmdlet retrieves all transport rules configured in the Exchange organization.
- It helps verify that the disclaimer rule ("Add Legal Disclaimer") is listed and enabled.
- The output displays the rule's name, status (Enabled/Disabled), mode (Enforce/Test), and priority.

Validation:
You should see an entry similar to:

Name	State	Mode	Priority
Add Legal Disclaimer	Enabled	Enforce	1

This confirms that the rule has been created and is currently active with a defined priority.

##
Step d) Remove the Block Large Messages Rule
##

Objective:
Delete the transport rule that blocks large messages to comply with updated message handling policies or avoid rule conflicts.

Action:
Remove the existing "Block Large Messages" rule from the Exchange organization.

Command:
Remove-TransportRule -Identity "Block Large Messages"

```
[PS] C:\Users\Administrator>Remove-TransportRule -Identity "Block Large Messages"

Confirm
Are you sure you want to perform this action?
Removing transport rule "Block Large Messages".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [?] Help (default is "Y"): A
```

Explanation:
- Remove-TransportRule: Deletes the specified transport rule.
- -Identity "Block Large Messages": Specifies the exact rule name to remove.
- You will be prompted to confirm the deletion unless `-Confirm:$false` is added for automation.

Optional for scripting:
Use the `-Confirm:$false` parameter to bypass the confirmation prompt during automation:

```
Remove-TransportRule -Identity "Block Large Messages" -Confirm:$false
```

```
##############################################################################
```
Step e) Verify that the Block Large Messages Rule Has Been Removed
```
##############################################################################
```

Objective:
Ensure that the "Block Large Messages" transport rule has been successfully deleted and is no longer active in the organization.

Action:
List all transport rules and check that "Block Large Messages" no longer appears.

Command:
```
Get-TransportRule | Format-Table Name,Priority,State
```

```
[PS] C:\Users\Administrator>Get-TransportRule | Format-Table Name,Priority,State

Name                Priority   State
----                --------   -----
Add Legal Disclaimer        0 Enabled
```

Explanation:
- Get-TransportRule: Retrieves all currently defined transport rules.
- Format-Table Name,Priority,State: Displays the rule name, priority, and whether it's enabled.
- This allows a quick visual check to confirm the rule is gone.

Optional:
For more detailed output, you can also use:

```
Get-TransportRule | Format-List Name,Comments,Priority,State
```

```
[PS] C:\Users\Administrator>Get-TransportRule | Format-List Name,Comments,Priority,State


Name     : Add Legal Disclaimer
Comments :
Priority : 0
State    : Enabled
```

##
Optional Commands: Exploring Transport Rule Components
##

Objective:
Understand how to retrieve detailed information about available conditions (predicates) and actions that can be used when defining transport rules.

1) **Command:**
Get-TransportRuleAction

Explanation:
- Lists all supported **actions** that can be applied in transport rules.
- Examples include: `RejectMessage`, `ApplyHtmlDisclaimer`, `RedirectMessageTo`, etc.
- Useful for **planning or scripting custom rules** beyond the basic GUI options.

2) **Command:**
Get-TransportRulePredicate

Explanation:
- Lists all available **conditions (predicates)** that transport rules can evaluate.
- Examples include: `FromScope`, `SentToScope`, `MessageSizeOver`, etc.
- Helpful for **understanding what filters you can apply** to target specific types of messages.

Use Case:
These commands are especially valuable when writing or reviewing **PowerShell-based transport rules** to ensure youre using valid syntax and supported conditions/actions. They also help expand your awareness of what's possible beyond what the EAC offers.