

# Transport agents: Exchange 2013 Help

By msdmaguire

Archived: 2026-04-06 01:28:05 UTC

***Applies to:*** Exchange Server 2013

Transport agents let you install custom software that is created by Microsoft, by third-party vendors, or by your organization, on an Exchange server. This software can then process email messages that pass through the transport pipeline. In Microsoft Exchange Server 2013, the transport pipeline is made of the following processes:

- The Front End Transport service on Client Access servers
- The Transport service on Mailbox servers
- The Mailbox Transport service on Mailbox servers
- The Transport service on Edge Transport servers

For more information about the transport pipeline, see [Mail flow](#)

Like the previous version of Exchange, Exchange 2013 transport provides extensibility through the Microsoft Exchange Server 2013 Transport Agents SDK. The Exchange 2013 version of the SDK is based on the Microsoft .NET Framework version 4.0 and allows third parties to implement the following predefined classes:

- **SmtptReceiveAgent**
- **RoutingAgent**
- **DeliveryAgent**

When compiled against libraries in the SDK, the resulting assemblies are registered with Exchange 2013, which loads the agents and invokes their event handlers during specific stages of the SMTP sessions or message processing. These stages, or events, are part of the agent definitions. The agent registration information is stored in an XML configuration file.

The following list explains the requirements for using transport agents in Exchange 2013.

- The Transport service on Mailbox servers and Edge Transport servers fully supports all the predefined classes in the SDK, and therefore any third-party transport agents written for the Hub Transport or Edge Transport server roles in Microsoft Exchange Server 2010 should work in the Transport service in Exchange 2013.
- The Front End Transport service only supports the **SmtptReceiveAgent** class in the SDK, and third-party agents can't operate on the **OnEndOfData** SMTP event.

- The Mailbox Transport service doesn't support the SDK at all, so you can't use any third-party agents in the Mailbox Transport service.

Support for legacy transport agents based on versions of the .NET Framework prior to version 4.0 isn't enabled by default, but you can enable it. For instructions, see [Enable support for legacy transport agents](#).

Due to the updates to the Exchange 2013 transport pipeline, the transport agent cmdlets need to distinguish between the Transport service and the Front End Transport service, especially if the Client Access server and the Mailbox server are installed on the same computer. For more information, see [Manage transport agents](#).

Transport Agent management cmdlets manipulate a configuration file located at `%ExchangeInstallPath%TransportRoles\Shared` . For the Transport service on Mailbox servers and Edge Transport servers, the file is `agents.config` . For the Front End Transport service on Client Access servers, the file is `fetagents.config` . Both files use the same format as in Exchange 2010. For more information about managing transport agents, see [Manage transport agents](#).

Transport agents use SMTP events. These events are triggered as messages move through the transport pipeline. SMTP events give transport agents access to messages at specific points during the SMTP conversation and during routing of messages through the organization.

Note that there are new SMTP Receive events in Exchange 2013. SMTP Receive exists in the Front End Transport service on Client Access servers, the Transport service on Mailbox servers and Edge Transport servers and the Mailbox Transport Delivery service on Mailbox servers. The categorizer exists only in the Transport service on Mailbox servers and Edge Transport servers. For more information about transport services and the categorizer, see [Mail routing](#).

The following tables list the SMTP events that provide access to messages in the transport pipeline.

Sequence	SMTP event	Description
1	<b>OnConnectEvent</b>	This event is triggered by the initial connection from a remote SMTP host.
2	<b>OnHeloCommand</b>	This event is triggered when the <code>HELO</code> command is issued by the remote SMTP host.
3	<b>OnEhloCommand</b>	This event is triggered when the <code>EHLO</code> command is issued by the remote SMTP host.
4	<b>OnStartTlsCommand</b>	This event is triggered when the <code>STARTTLS</code> command is issued by the remote SMTP host.
5	<b>OnAuthCommand</b>	This event is triggered when the <code>AUTH</code> command is issued by the remote SMTP host.
6	<b>OnProcessAuthentication</b>	This event is triggered when authentication with the remote SMTP host is being processed.

Sequence	SMTP event	Description
7	<b>OnEndOfAuthentication</b>	This event is triggered when the remote SMTP host has completed authentication.
8	<b>OnXSessionParamsCommand</b>	This event is triggered when the <code>XSESSIONPARAMS</code> command is issued by the remote SMTP host.
9	<b>OnMailCommand</b>	This event is triggered when the <code>MAIL FROM</code> command is issued by the remote SMTP host.
10	<b>OnRcptToCommand</b>	This event is triggered when the <code>RCPT TO</code> command is issued by the remote SMTP host.
11	<b>OnDataCommand</b>	This event is triggered when the <code>DATA</code> (text) or <code>BDAT</code> (binary data) command is issued by the remote SMTP host.
12	<b>OnEndOfHeaders</b>	This event is triggered when the remote SMTP host has completed submitting the email message headers. This is indicated by a blank line ( <code>&lt;CRLF&gt;</code> ) that separates the message headers and the message body.
13	<b>OnProxyInboundMessage</b>	This event is triggered when an inbound SMTP session is relayed or <i>proxied</i> by the Front End Transport service on a Client Access server to the Transport service on a Mailbox server.
14	<b>OnEndOfData</b>	This event is triggered when the remote SMTP host issues an end of data command. For text sessions started by the <code>DATA</code> command, the end of data indicator is <code>&lt;CRLF&gt;</code> . <code>&lt;CRLF&gt;</code> . For binary sessions started by the <code>BDAT</code> command, the end of data indicator is <code>BDAT LAST</code> .
**	<b>OnHelpCommand</b>	This event is triggered if the <code>HELP</code> command is issued by the remote SMTP host.
**	<b>OnNoopCommand</b>	This event is triggered if the <code>NOOP</code> command is issued by the remote SMTP host.
**	<b>OnReject</b>	This event is triggered if the receiving SMTP host issues a temporary or permanent delivery status notification (DSN) code to the sending SMTP host.
**	<b>OnRsetCommand</b>	This event is triggered if the <code>RSET</code> command is issued by the sending SMTP host.

Sequence	SMTP event	Description
15	<b>OnDisconnectEvent</b>	This event is triggered by the disconnection of the SMTP conversation by either the receiving or sending SMTP host. Typically, this happens when the <code>QUIT</code> command is issued by the remote SMTP host.

\*\* These events can occur at any time after **OnConnectEvent** but before **OnDisconnectEvent**.

Sequence	SMTP event	Description
1	<b>OnSubmittedMessage</b>	This event is triggered when a message arrives in the Submission queue in the Transport service on the receiving Mailbox server or Edge Transport server.
2	<b>OnResolvedMessage</b>	This event is triggered after all the recipients have been resolved, but before the next hop has been determined for each recipient. The <b>OnResolvedMessage</b> routing event enables subsequent events to override the default routing behavior by using the per-recipient <b>SetRoutingOverride</b> method.
3	<b>OnRoutedMessage</b>	This event is triggered after messages have been categorized, distribution lists have been expanded, and recipients have been resolved.
4	<b>OnCategorizedMessage</b>	This event is triggered when the categorizer completes processing the message.

There are two factors that determine the order that transport agents act on messages in the transport pipeline:

1. The SMTP event where the transport agent is registered, and when that SMTP event encounters messages.
2. The priority value that's assigned to the transport agent if there are multiple agents registered to the same SMTP event. The highest priority is 1. A higher integer value indicates a lower agent priority.

For example, suppose you configured the following transport agents:

- Transport Agent A with a priority of 1 and Transport Agent C with a priority of 2 are registered to the **OnEndOfHeaders** SMTP event.
- Transport Agent B with a priority of 4 is registered to the **OnMailCommand** SMTP event.

Transport Agent B is applied to messages first because the **OnMailCommand** event encounters messages before the **OnEndOfHeaders** event. When messages reach the **OnEndOfHeaders** event, Transport Agent A is applied before Transport Agent C because Transport Agent A has a higher priority (lower integer value) than Transport Agent C.

Exchange 2013 includes many built-in transport agents that provide features such as anti-spam, transport rules and journaling. Most of the built-in transport agents on Exchange 2013 Mailbox servers and Client Access servers are invisible and unmanageable by the transport agent management cmdlets. Virtually all of the built-in transport agents that are visible and manageable are in the Transport service on Mailbox servers and on Edge Transport servers.

The more interesting built-in transport agents on Mailbox servers are described in the following table. Note that this table doesn't include many of the invisible and unmanageable transport agents.

<b>Agent name</b>	<b>Manageable?</b>	<b>Priority</b>	<b>SMTP or categorizer events</b>
Transport Rule Agent	Yes	1	<b>OnResolvedMessage</b>
Malware Agent	Yes	2	<b>OnSubmittedMessage</b>
Text Messaging Routing Agent	Yes	3	<b>OnSubmittedMessage</b>
Text Messaging Delivery Agent	Yes	4	n/a
Journal Agent	No	Not configurable	<b>OnRoutedMessage</b>
Journal Report Decryption Agent	No	Not configurable	<b>OnCategorizedMessage</b>
RMS Decryption Agent	No	Not configurable	<b>OnSubmittedMessage</b>
RMS Encryption Agent	No	Not configurable	<b>OnSubmittedMessage</b> <b>OnRoutedMessage</b>
RMS Protocol Decryption Agent	No	Not configurable	<b>OnEndOfData</b>

On Edge Transport servers, most of the built-in transport agents are visible and manageable by the transport agent management cmdlets or by other feature-specific cmdlets.

The more interesting built-in transport agents on Edge Transport servers are described in the following table. Note that this table doesn't include invisible or unmanageable transport agents.

<b>Agent name</b>	<b>Manageable?</b>	<b>Priority</b>	<b>SMTP or categorizer events</b>
Connection Filtering Agent	Yes	1	<b>OnConnectEvent</b> <b>OnMailCommand</b> <b>OnRcptComand</b> <b>OnEndOfHeaders</b>
Address Rewriting Inbound Agent	Yes	2	<b>OnRcptComand</b> <b>OnEndOfHeaders</b>
Edge Rule Agent	Yes	3	<b>OnEndOfData</b>

Agent name	Manageable?	Priority	SMTP or categorizer events
Content Filter Agent *	Yes	4	<b>OnEndOfData</b>
Sender ID Agent *	Yes	5	<b>OnEndOfHeaders</b>
Sender Filter Agent *	Yes	6	<b>OnMailCommand</b> <b>OnEndOfHeaders</b>
Recipient Filter Agent	Yes	7	<b>OnRcptCommand</b>
Protocol Analysis Agent *	Yes	8	<b>OnConnectEvent</b> <b>OnEndOfHeaders</b> <b>OnEndOfData</b> <b>OnReject</b> <b>OnRsetCommand</b> <b>OnDisconnectEvent</b>
Attachment Filtering Agent	Yes	9	<b>OnEndOfData</b>
Address Rewriting Outbound Agent	Yes	10	<b>OnSubmittedMessage</b> <b>OnRoutedMessage</b>

\* You can also install and configure these anti-spam agents on Mailbox servers. For more information, see [Enable anti-spam functionality on Mailbox servers](#).

To help you troubleshoot issues with transport agents, you can use the following features:

- **Get-TransportPipeline:** This cmdlet shows the SMTP events and the corresponding transport agents that encounter messages on the Exchange server. For more information, see [View transport agents in the transport pipeline](#).
- **Pipeline Tracing:** Pipeline tracing creates an exact snapshot of a message before and after it encounters each transport agent. This allows you to find a transport agent that's causing unexpected results. For more information, see [Pipeline tracing](#).

---

Source: <https://docs.microsoft.com/en-us/exchange/transport-agents-exchange-2013-help>