Microsoft BizTalk Server 2006 Part-VII

Table of Contents

Module 42: Configuring the HTTP Adapter
- How to Configure an HTTP Receive Handler
- How to Configure IIS for an HTTP Receive Location
- How to Configure an HTTP Receive Location
- How to Configure an HTTP Send Handler
- Configuring an HTTP Send Port
- HTTP Adapter Configuration and Tuning Parameters
- HTTP Adapter Property Schema and Properties

Module 43: Configuring an HTTP Send Port
- How to Configure an HTTP Send Port
- How to Configure an HTTP Send Port With a Remote BizTalk Management Database
- Restrictions on the Destination URL Property

Module 44: MQSeries Adapter
- What Is the MQSeries Adapter?
- Configuring the MQSeries Adapter
- Walkthrough: Creating a BizTalk Application That Uses the MQSeries Adapter
- MQSeries Adapter Batching and Transaction Handling
- Correlating Messages Using Request-Reply
- MQSeries Adapter Custom Headers
- Analyzing MQSeries Adapter Errors with the Trace Tools
- Ordered Delivery of Messages with the MQSeries Adapter

Module 45: What Is the MQSeries Adapter?
- Components of the MQSeries Adapter
- MQSeries Adapter Architecture
- Using the MQSeries Adapter
- MQSeries Adapter Security

Module 46: MQSeries Adapter Architecture
- Structure of the MQSeries Adapter
- MQSeries Queues

Module 47: Using the MQSeries Adapter
- MQSeries Adapter Deployment Options
- MQSeries Adapter Message Flow
- Using MQSeries Adapter with an Earlier Version of the Adapter
- MQSeries Adapter High Availability

Module 48: Configuring the MQSeries Adapter
- How to Configure MQSeries Adapter Receive Locations and Send Ports
- How to Configure MQSeries Adapter Send and Receive Handlers
- Using the MQSAgent COM+ Configuration Wizard
- Silent Configuration of the MQSeries Adapter
- MQSeries Adapter Properties

Module 49: Silent Configuration of the MQSeries Adapter
- Command-Line Configuration Wizard for the MQSeries Adapter
- XML Configuration File for the MQSeries Adapter

Module 50: MQSeries Adapter Properties
- Data Type Conversion of Properties
- Properties Related to BizTalk Server
- MQSeries Context Properties

Module 51: MSMQ Adapter
- What Is the MSMQ Adapter?
- Configuring the MSMQ Adapter
- Reliable Messaging with the MSMQ Adapter
- Analyzing MSMQ Adapter Errors with the Trace Tool
- MSMQ Adapter Property Schema and Properties
- Ordered Delivery of Messages with the MSMQ Adapter

Module 52: What Is the MSMQ Adapter?
- MSMQ Adapter Architecture

Module 53: Configuring the MSMQ Adapter
- How to Configure an MSMQ Receive Handler
- How to Configure an MSMQ Receive Location
- How to Configure an MSMQ Send Handler
- How to Configure an MSMQ Send Port
- Configuring the MSMQ Adapter Properties
- How to Manage Multiple Receive Locations
- Message Queuing Queues
- Optimizing Performance of the MSMQ Adapter
- Sending and Receiving Large Messages
- Setting Up the MSMQT and MSMQ Adapters on One Computer

Module 54: Reliable Messaging with the MSMQ Adapter
- Properties for Reliable Messaging with the MSMQ Adapter
- Transaction Handling with the MSMQ Adapter

Module 55: POP3 Adapter
- What Is the POP3 Adapter?
- Configuring the POP3 Adapter
- Walkthrough: Creating a BizTalk Application That Uses the POP3 Adapter

Module 56: Configuring the POP3 Adapter
- How to Configure a POP3 Receive Handler
- How to Configure a POP3 Receive Location
- POP3 Adapter Property Schema and Properties

Module 57: SMTP Adapter
- Configuring the SMTP Adapter
- Restrictions When Configuring the SMTP Adapter
- SMTP Adapter Security Recommendations

Module 58: Configuring the SMTP Adapter
- How to Configure an SMTP Send Handler
- How to Configure an SMTP Send Port

How to Remove Other Files and Settings for a BizTalk Application

This topic describes how to remove files and settings for a BizTalk application that may not be removed when you uninstall the application (which is described in How to Uninstall a BizTalk Application). For example, certificates, COM and COM+ registry entries, and COM files are not removed unless the application included a script to remove them on uninstallation.

- **Delete Certificates.** There are two ways to delete certificates from the certificate store: by using the Certificate Manager (certmgr.exe) command-line tool or the Certificates snap-in. Certmgr.exe is installed with the .NET SDK, which is one of the BizTalk Server 2006 installation prerequisites. You can run certmgr.exe manually, or you can run it from a post-processing script that runs following application uninstallation. For more information about using certmgr.exe, see Certificate Manager Tool (certmgr.exe) at the Microsoft Web site.

  The Certificates snap-in is included in both Windows Server 2003 and Windows XP Professional. To delete a certificate, open the snap-in, as described in "Starting the Certificates snap-in" in Help for your operating system, and then delete the certificate as described in "Delete a certificate" in Certificates Help.

- **Remove assemblies from the Windows Registry.** To remove .NET and BizTalk assemblies from the Windows registry, use regsvcs or regasm, which are included with the .NET SDK, which is one of the BizTalk Server 2006 installation prerequisites. For reference information, see .NET Services Installation Tool (Regsvcs.exe) and Assembly Registration Tool (Regasm.exe) at the Microsoft Web site.

- **Remove COM components from the Windows Registry.** To remove COM components from the Windows Registry, use regsvr32. For reference information, see "Regsvr32" in Help for your operating system. This tool is included in both Windows Server 2003 and Windows XP Professional.

- **Uninstall assemblies from the global assembly cache (GAC).** Assemblies are not automatically uninstalled from the GAC. Generally, this is necessary only in a development environment. You can uninstall an assembly from the GAC manually or by using a script. For more information, see How to Uninstall an Assembly from the GAC.

Prerequisites

To remove the files and settings described in this topic, you must be logged on with Write permissions on the Windows Registry or the certificate store, depending on what you want to remove. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

Managing Parties

This section provides information about how to manage parties (entities outside of BizTalk Server that interact with an orchestration) in a BizTalk Server group. Using the **Parties** node, you can set up business partners or internal departments with which BizTalk Server solutions interact.

In This Section

- How to Create a Party
- How to Delete a Party
- How to View or Edit a Party

How to Create a Party

You can create and configure party using the BizTalk Server Administration Console. A party is an entity outside of BizTalk Server that interacts with an orchestration. All of the partners that your organization deals with are considered parties, and your organization may have several thousand partners. You can use the Administration Console to manage parties, but managing thousands of parties can be more easily handled using the BizTalk Explorer Object Model. Using the BizTalk Explorer Object Model, you can write code to add parties, remove parties, add a party and automatically enlist to a role, or copy a party from a legacy system.

After you create a party in BizTalk Server, it is likely that you want to interact with that party using an orchestration. Parties interact with orchestrations through roles.

For example, you might have a shipping role in your orchestration. The shipper would have one or more parties associated with it. When the orchestration needs to decide the least expensive shipping company to use to ship an item, it can compare the prices of the parties in the shipper role.

You must enlist a party to tie it to a specific role. Enlisting a party enables an orchestration to interact with the party. For more information, see How to Enlist or Unenlist a Party for a Role.

Prerequisites

To perform this procedure, you must be logged on as a member of the BizTalk Server Administrators group.

To create a party

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, right-click **Parties**, click **New**, and then click **Party**.

3. In the **Party Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Name	Type the party name or select it from the drop-down list.
Delete	Click to delete the selected alias. (An alias is a unique identifier for a party; a party can have more than one alias.)
Aliases - Name	From the drop-down list, select the name of the organization for which you want to supply an alias.
Aliases - Qualifier	Type a qualifier for the selected organization. The qualifier distinguishes different aliases with the same value. The qualifier-value pair is unique for the organization.
Aliases - Value	Type the new alias for the selected organization.

4. On the **Send Ports** tab, do the following:

Use this	To do this
Remove	Click to remove the selected send port from the party.
Properties	Click to display the **Send Port Properties** window for the selected send port.
Send ports - Name	From the drop-down list, select a port to bind to the party.
Send ports - URI	Displays the send port address.

5. On the **Certificate** tab, do the following, and then click **OK**:

Use this	To do this
Signature certificate - Description	Displays a description of the selected certificate.
Signature certificate - Thumbprint	Displays the thumbprint of the private key certificate that is used for party resolution and signature verification. The certificate thumbprint has the format HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH, where H is a hexadecimal digit (a number from 0 through 9 or a letter from A through F).
Remove certificate	Click to remove the displayed certificate.

Browse	Click to display the **Select Certificate** dialog box, where you select the signature certificate from the Local Machine or Other People certificate store to apply to messages transmitted to the party.

How to Delete a Party

When your organization no longer needs to interact with a party, you can delete the party using the BizTalk Server Administration Console. A party is an entity outside of BizTalk Server that interacts with an orchestration.

Prerequisites

To perform this procedure, you must be logged on as a member of the BizTalk Server Administrators group.

To delete a party

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, and then click **Parties**.
3. In the details pane, right-click the party you want to delete, and then click **Delete**.
4. In the **Confirm delete party** dialog box, click **Yes** to delete the party.

How to View or Edit a Party

You can view or edit a party using the BizTalk Server Administration Console. A party is an entity outside of BizTalk Server that interacts with an orchestration. After a party is enlisted to a role and the ports are bound, you can:

- Edit the party's name and description.
- Add any new send ports to the party.
- Edit the party's certificate.

You cannot, however, remove the send port from a party that is bound to a role.

Prerequisites

To perform this procedure, you must be logged on as a member of the BizTalk Server Administrators group.

To view or edit a party

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, and then click **Parties**.

3. In the details pane, right-click the party you want to view or edit, and then click **Properties**.

4. In the **Party Properties** dialog box, view or edit the party as appropriate.

5. On the **General** tab, do the following:

Use this	To do this
Name	Type the party name or select it from the drop-down list.
Delete	Click to delete the selected alias.
Aliases - Name	From the drop-down list, select the name of the organization for which you want to supply an alias.
Aliases - Qualifier	Type a qualifier for the selected organization. The qualifier distinguishes different aliases with the same value. The qualifier-value pair is unique for the organization.
Aliases - Value	Type the new alias for the selected organization.

6. On the **Send Ports** tab, do the following:

Use this	To do this
Remove	Click to delete the selected send port from the party.
Properties	Click to display the **Send Port Properties** window for the selected send port.
Send ports - Name	From the drop-down list, select a port to bind to the party.
Send ports - URI	Displays the send port address.

7. On the **Certificate** tab, do the following, and then click **OK**:

Use this	To do this
Signature certificate –	Displays a description of the selected certificate.

Description	
Signature certificate Thumbprint -	Displays the thumbprint of the private key certificate that is used for party resolution and signature verification. The certificate thumbprint has the format HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH, where H is a hexadecimal digit (a number from 0 through 9 or a letter from A through F).
Remove certificate	Click to remove the displayed certificate.
Browse	Click to display the **Select Certificate** dialog box, where you select the signature certificate from the Local Machine or Other People certificate store to apply to messages transmitted to the party.

Managing Platform Settings

The topics in this section describe how to manage hosts, host instances, servers, MessageBox databases, and how to use adapters.

In This Section

- Managing BizTalk Hosts and Host Instances
- Managing Servers
- Managing MessageBox Databases
- Using Adapters

Managing BizTalk Hosts and Host Instances

A BizTalk Server Host is a logical set of zero or more BizTalk Server run-time processes in which you deploy items such as adapter handlers, receive locations (including pipelines), and orchestrations.

A host instance is the place where the message processing, receiving, and transmitting occurs. You install a host instance on each server running BizTalk Server 2006 that has one or more hosts mapped to that server.

Hosts have the following characteristics:

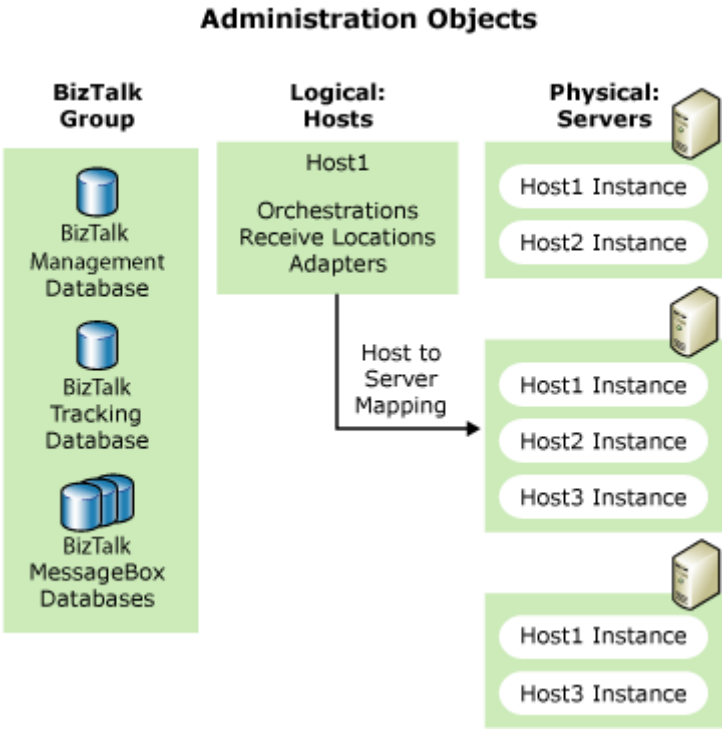
- Hosts are the logical containers of BizTalk Server objects.
- Only one instance of a specific host can exist on each server.
- You can map one host to multiple servers.

Host instances have the following characteristics:

- Host instances are the physical containers of BizTalk Server objects.
- You create a host instance when you map a server to a host.
- Multiple host instances (of different hosts) can exist on a server.

The following figure shows the relationship between servers, hosts, and host instances.

Relationship between hosts, host instances, and servers



In This Section

- How to Create a BizTalk Server Hosting Environment
- How to Create a New Host
- How to Modify Host Properties
- How to Remove a Host Group from a SQL Server Role
- How to Delete a Host
- How to Add a Host Instance
- How to Start a Host Instance

- How to Stop a Host Instance
- How to Delete a Host Instance
- How to Modify Host Instance Properties

How to Create a BizTalk Server Hosting Environment

Before you create your BizTalk Server hosting environment, consider the following recommendations:

- **Use different hosts for orchestrations and receive handlers**

  Any items running in a host (for example, orchestrations, pipelines, receive and send handlers) run under the same identity, and have access to the work and suspended queues for that host.

  If a message cannot be delivered to an orchestration due to permission errors, the message is placed in the suspended queue of the host where the sending process (a receive pipeline or another orchestration) is running. However, if the orchestration and the sending process (for example, a receive pipeline) are running in the same host, the orchestration can still access the message in the suspended queue. This could potentially compromise your system if a non-trusted orchestration is running in a trusted host.

  We recommend that you run non-trusted orchestrations in a separate host, with a different service account than the trusted hosts in your BizTalk group. For information about designating a host as trusted, see How to Modify Host Properties.

- **Limit the database and log size in the BizTalk Server databases**

  The BizTalk MessageBox databases and the BizTalk Tracking database grow much faster than the other BizTalk Server databases. As part of your backup and maintenance program, you should update these databases frequently.

  By default, the tables in the BizTalk Server databases do not have a log size limit. As part of your backup and maintenance program, we recommend that you limit the log size to prevent the logs from becoming too large and potentially using all the disk space.

- **Use SQL Server clustering**

  To provide high availability of the BizTalk Server databases, we recommend that you cluster the SQL servers where the BizTalk Server databases are stored. This will help minimize downtime if one of the databases or SQL Server fails. For more information about SQL Server clustering, see "Failover Clustering Architecture" in SQL Server 2000 Books Online.

Prerequisites

The following are prerequisites for performing the procedure in this topic:

- You must be logged on as a member of the BizTalk Server Administrators group.
- The instructions in the following procedure assume that you have installed BizTalk Server with the complete installation option. If you did not install BizTalk Server with the complete installation option, some of the administration objects listed in step 1 may not be on your system.

To create a BizTalk Server hosting environment

1. Use the Configuration Wizard to create a new BizTalk group. For information about creating a new BizTalk group, see Working with the Configuration Framework.

   The Configuration Wizard creates the following administration objects:

Administration object	Description
BizTalk Management database (BizTalkMgmtDb)	This database is the central meta-information store for all BizTalk servers.
BizTalk MessageBox database (BizTalkMsgBoxDb)	This database stores subscriptions predicates. It is a host platform, and keeps the queues and state tables for each BizTalk Server host. The MessageBox database also stores the messages and message properties. For information about MessageBox databases, including adding additional MessageBox databases, see Managing MessageBox Databases.
Server	This is the computer on which BizTalk Server is installed and configured, and where host instances are running. You create host instances from a host created on a server. For more information about creating a host, see How to Create a New Host. For information about creating host instances, see How to Add a Host Instance.
BAM Primary Import database (BAMPrimaryImport)	This is the database where the Business Activity Monitoring tool collects tracking data.
Rule Engine database (BizTalkRuleEngineDb)	This database is a repository for policies, rules, and vocabularies for data references in business rules.
BizTalk Tracking database (BizTalkDTADb)	This database stores business and health-monitoring data tracked by the BizTalk Server tracking engine.
SSO database (SSODB)	This database stores credential information.
Tracking Analysis Server Administration database,	This database stores both business and health-monitoring

named BizTalkAnalysisDb	OLAP cubes.
In-process host with corresponding host instances	The in-process host operates within the BizTalk Server process space.
Isolated host with corresponding host instances	The isolated host operates outside of the BizTalk Server installation.
HTTP/S, BizTalk Message Queuing, File, SMTP, SOAP, and SQL	The Configuration Wizard creates the adapters that are part of BizTalk Server 2006.

2. Use the BizTalk Server 2006 Administration Console or WMI to add components to your BizTalk Server environment as needed. To scale out your solution, add MessageBox databases, hosts, and servers.

3. Use the BizTalk Administration Console or WMI to create host instances on the mapped servers. This step determines on which servers BizTalk Server will run. As the needs in your enterprise change, you can add servers, remove servers, and change server-to-host mapping.

How to Create a New Host

A BizTalk Host is a logical container for items such as adapter handlers, receive locations (including pipelines), and orchestrations. We recommend that you use separate hosts for processing, receiving, and sending messages, and that you use separate hosts for trusted and non-trusted items to facilitate implementing security measures and to improve manageability of the hosts. You can install only one host per BizTalk server.

For information about using Windows Management Instrumentation (WMI) to create a new host, see **MSBTS_Host (WMI)** .

Prerequisites

You must have the following user rights to create hosts, modify host properties, and delete hosts:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Server Administrators group, see Managing the BizTalk Administrators Group .

- You must have the following rights in SQL Server:

  - You must be either a SQL Server administrator, or a member of the db_owner or db_securityadmin SQL Server database roles in the BizTalk Tracking database (BizTalk DTADb), MessageBox databases (BizTalkMsgBoxDb), and the BAM Primary Import database (BAMPrimaryImport).

- You must be a member of the sysadmin SQL Server role on all the computers where there are MessageBox databases, or a member of the db_owner or db_ddladmin SQL Server role for all the MessageBox databases.

To create a new host

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.

3. In the details pane, right click **Hosts**, click **New**, and then click **Host**.

4. In the **Host Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Name	Displays the name of the host. You name the host when you create it. The host name cannot exceed 80 characters in length.
Type	Displays the host type. You can enlist an orchestration to an In-process host, and an In-process host can host a send or receive handler. An In-process Host can host orchestrations, certain send handlers and receive handlers. An Isolated Host is a process boundary outside of BizTalk Server and is required by certain send and receive handlers
Options - Allow Host Tracking	Select this check box to indicate that the host loads the BizTalk Tracking component to process health monitoring and business data. If you select this check box, the current host will have read/write privileges to the tracking tables in the MessageBox database, as well as to the BizTalk Tracking database. Accordingly, any objects running in this host will also have read/write access to these databases. If you clear the check box, the host will have only write access to the tracking tables in the MessageBox database and will not have access to the BizTalk Tracking database.
Options - Authentication Trusted	Select this check box to specify that BizTalk Server should trust this host.
Options - 32-bit only	Select this check box if you want the host instance process to be created as 32-bit on both 32-bit and 64-bit servers. If this check box is cleared, the host instance process will be created as 32-bit on 32-bit servers and as 64-bit on 64-bit servers.
Options - Make this the default host in the group	Select this check box to identify this host as the default host. The orchestration binding process automatically uses the default host to host the orchestration, unless you explicitly select a different host. If

	this check box is selected and unavailable, this host is the default.
Windows group	Type the local or domain group for the host. Members of the Windows group will have permissions to run instances of this host.

5. On the **Advanced** tab, do the following:

Use this	To do this
Maximum number of messaging engine threads per CPU	Type or select the maximum number of threads per CPU that should be created for the messaging engine to process the workload. This number does not include the threads created by the individual adapters or the NT thread-pool.
Throttling Thresholds	Click to display the **Throttling Thresholds** dialog box.
Message Publishing Throttling	Click to display the **Message Publishing Throttling Settings** dialog box.
Message Processing Throttling	Click to display the **Message Processing Throttling Settings** dialog box.
Restore Defaults	Click to restore default settings in the **Maximum number of messaging engine threads per CPU** check box and all three **Throttling** dialog boxes.

6. On the **Certificates** tab, do the following, and then click **OK**:

Use this	To do this
Decryption certificate Description -	Displays a description of the displayed decryption certificate.
Decryption certificate Thumbprint -	Displays the thumbprint of the private key certificate used to decrypt inbound messages for this host. The certificate thumbprint has the format HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH, where H is a hexadecimal digit (a number from 0 through 9 or a letter from A through F).
Decryption certificate Remove certificate -	Click to remove the displayed decryption certificate from the host.
Decryption certificate Browse -	Click to display the **Select Certificate** dialog box, where you select the decryption certificate from the Local Machine or Other People certificate store that you want to use with the host.

How to Modify Host Properties

You can use the BizTalk Server 2006 Administration Console to modify the following host properties:

- **Windows group:** Members of the Windows group have permissions to run the hosts instances of this host by default. When you select a Windows user group for a BizTalk Host, we recommend that you create a new group that is dedicated to that host. If you do use an existing group, ensure that the group does not have more privileges than are needed. For more information, see **Access Control and Data Security**.

- **Host tracking:** At least one host in the group must track health and business data. This option indicates whether the host loads the BizTalk Tracking component to process health monitoring and business data.

- The host that performs host tracking has read/write access to the tracking tables in the MessageBox database, as well as access to the BizTalk Tracking database. Therefore, any objects running in a host that performs host tracking also have read/write access to these database tables. Hosts that do not perform host tracking have only write access to the tracking tables in the MessageBox database, and do not have access to the BizTalk Tracking database.

  Any host marked as "host tracking" no longer uses the BizTalk Host group. BizTalk Administration automatically removes the BizTalk Host group from the SQL Server roles for the BizTalk Tracking database. You must manually remove the BizTalk Host group from the SQL Server roles for the BizTalk Management database and the MessageBox database. For information about removing a BizTalk Host group from SQL Server roles, see How to Remove a Host Group from a SQL Server Role.

- **Authentication trusted:** You can specify that BizTalk Server trusts a host. BizTalk Server trusts **authentication trusted hosts** to place the sender security ID (SSID) on messages that the trusted host is queuing that map to users other than to the host. For more information about authentication trusted hosts, see **Authenticating the Sender of a Message**.

  Host instances of trusted hosts and host instances of non-trusted hosts cannot use the same service accounts. If you want to change the trust setting of a host instance and the host instance uses a service account that other host instances use, you can do one of the following:

  - You can change the service account of the host instance for which you want to change the trust settings to a new service account.

  - You can change the service account of the host instance to an existing service account that other host instances with the same trust setting use.

  - You can delete the host instance, and re-create it with a different trust setting and service account.

- **Default host in the group:** There must be a default host in the group at all times. The orchestration enlistment process automatically uses the default host to host the orchestration, unless the user explicitly selects a different host. The first host created is marked as the default host. For information about the default host, see Hosts.

Prerequisites

You must have the following user rights to create hosts, modify host properties, and delete hosts:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Server Administrators group, see Managing the BizTalk Administrators Group .

- You must have the following rights in SQL Server:

  - You must be either a SQL Server administrator, or a member of the db_owner or db_securityadmin SQL Server database roles in the BizTalk Tracking database (BizTalk DTADb), MessageBox databases (BizTalkMsgBoxDb), and the BAM Primary Import database (BAMPrimaryImport).

  - You must be a member of the sysadmin SQL Server role on all the computers where there are MessageBox databases, or a member of the db_owner or db_ddladmin SQL Server role for all the MessageBox databases.

To modify host properties

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.

3. In the details pane, right click the host you want to modify and then click **Properties**.

4. In the **Host Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Name	Displays the name of the host. You name the host when you create it.
Type	Displays the host type. You can enlist an orchestration to an In-process host, and an In-process host can host a send handler. An Isolated host operates outside the BizTalk Server installation.
Options - Allow Host Tracking	Select this check box to indicate that the host loads the BizTalk Tracking component to process health monitoring and business data. If you select this check box, the current host will have read/write access to the tracking tables in the MessageBox database, as well as to the Tracking database. Accordingly, any objects running in this host will also have read/write access to these databases. If you clear

	the check box, the host will have only write access to the tracking tables in the MessageBox database and will not have access to the Tracking database.
Options - Authentication Trusted	Select this check box to specify that BizTalk Server should trust this host.
Options - 32-bit only	Select this check box if you want the host instance process to be created as 32-bit on both 32-bit and 64-bit servers. If this check box is cleared, the host instance process will be created as 32-bit on 32-bit servers and as 64-bit on 64-bit servers.
Options - Make this the default host in the group	Select this check box to identify this host as the default host. The orchestration enlistment process automatically uses the default host to host the orchestration, unless you explicitly select a different host. If this check box is selected and unavailable, this host is the default.
Windows group	Type the local or domain group for the host. Members of the Windows group will have permissions to run instances of this host.

5. On the **Advanced** tab, do the following:

Use this	To do this
Maximum number of messaging engine threads per CPU	Type or select the maximum number of threads per CPU that should be created for the messaging engine to process the workload. This number does not include the threads created by the individual adapters or the NT thread-pool.
Throttling Thresholds	Click to display the **Throttling Thresholds** dialog box.
Message Publishing Throttling	Click to display the **Message Publishing Throttling Settings** dialog box.
Message Processing Throttling	Click to display the **Message Processing Throttling Settings** dialog box.
Restore Defaults	Click to restore default settings in the **Maximum number of messaging engine threads per CPU** check box and all three **Throttling** dialog boxes.

6. On the **Certificates** tab, do the following, and then click **OK**:

Use this	To do this
Decryption certificate Description -	Displays a description of the displayed decryption certificate.
Decryption certificate Thumbprint -	Displays the thumbprint of the private key certificate used to decrypt inbound messages for this host. The certificate thumbprint has the format HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH, where H is a hexadecimal digit (a number from 0 through 9 or a letter from A through F).
Decryption certificate Remove certificate -	Click to remove the displayed decryption certificate from the host.
Decryption certificate Browse -	Click to display the **Select Certificate** dialog box, where you select the decryption certificate from the Local Machine or Other People certificate store that you want to use with the host.

How to Remove a Host Group from a SQL Server Role

BizTalk Server grants the BizTalk Host user group permissions to run stored procedures in the BizTalk Management database, MessageBox database, and BizTalk Tracking database by means of SQL Server role memberships.

Any host marked as "host tracking" no longer uses the BizTalk Host group. The BizTalk Server 2006 Administration Console automatically removes the BizTalk Host group from the SQL Server roles for the BizTalk Tracking database.

You must manually remove the BizTalk Host group from the SQL Server roles for the BizTalk Management database and the MessageBox database. For information about the host tracking property, see How to Modify Host Properties.

Prerequisites

You must be a SQL Server system administrator to remove users from a SQL Server role group.

To remove a host group from a SQL Server role

1. Open SQL Server Enterprise Manager. Click **Start**, click **Run**, type **"SQL Server Enterprise Manager.msc"** and then click **OK**.

2. Open the appropriate server by clicking it, click **Databases**, click **BizTalkMgmtDb**, and then click **Roles**.

3. In the details pane, right-click **BTS_HOST_USERS**, and then click **Properties**.

4. In the **Database Role Properties** dialog box, select the BizTalk Host group you want to remove, click **Remove**, and then click **OK**.

5. In the console tree, click **BizTalkMsgBoxDb**, and then click **Roles**.

6. In the details pane, right-click **BTS_HOST_USERS**, and then click **Properties**.

7. In the **Database Role Properties** dialog box, select the BizTalk Host group you want to remove, click **Remove**, and then click **OK**.

How to Delete a Host

You can delete a host only in the following circumstances:

- It is not the default host.
- It is not the only host that is performing host tracking.
- It is not hosting any adapter handlers.
- It has no enlisted orchestrations.
- There are no started instances of the host.

Prerequisites

You must have the following user rights to create hosts, modify host properties, and delete hosts:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Server Administrators group, see Managing the BizTalk Administrators Group .
- You must have the following rights in SQL Server:
  - You must be either a SQL Server administrator, or a member of the db_owner or db_securityadmin SQL Server database roles in the BizTalk Tracking database (BizTalk DTADb), MessageBox databases (BizTalkMsgBoxDb), and the BAM Primary Import database (BAMPrimaryImport).
  - You must be a member of the sysadmin SQL Server role on all the computers where there are MessageBox databases, or a member of the db_owner or db_ddladmin SQL Server role for all the MessageBox databases.

To delete a host

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.

3. In the details pane, right click the host you want to delete and then click **Delete**.

4. In the **Confirm host delete** dialog box, click **Yes**.

How to Add a Host Instance

You can use the BizTalk Server 2006 Administration Console or Windows Management Instrumentation (WMI) to add host instances. BizTalk Server 2006 enabled you to add a host instance to all servers at the same time. In BizTalk Server 2006, however, you can only add a host instance to one server at a time. For information about using WMI to add a host instance, see **MSBTS_HostInstance (WMI)** .

Adding a host instance maps the instance of a given host to an instance of BizTalk Server. If you have an existing host instance that you must repair, you can attempt to repair the host instance by adding it again to the instance of BizTalk Server to which it is mapped. You must stop an existing host instance before you can add it again. For information about stopping a host instance, see How to Stop a Host Instance.

Prerequisites

You must have the following user rights to create host instances, modify host instance properties, start a host instance, stop a host instance, and delete host instances:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Administrators group, see Managing the BizTalk Administrators Group .
- You must be a member of the Administrators group on the run-time computer.
- You must be a SQL Server administrator (a member of the sysadmin SQL Server role) on the computer running SQL Server.
- You must be a member of the db_accessadmin and db_securityadmin SQL Server database roles for the following databases:
    - BizTalk MessageBox (BizTalkMsgBoxDb) (all)
    - BizTalk Tracking (BizTalk DTADb)
    - Rule Engine (BizTalkRuleEngineDb)
    - BizTalk Management (BizTalkMgmtDb)
    - BAM Primary Import (BAMPrimaryImport)

To add a host instance

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.

3. In the details pane, right click the host you want to create a new instance of, click **New**, and then click **Host Instance**.

4. In the **Host Instance Properties** dialog box, do the following, and then click **OK**:

Use this	To do this
Host name	Displays the name of the host associated with the selected server.
Server	Displays the server associated with the selected host.
Logon	Displays the account name of the new service account under which the host instance will run.
Configure	Click to display the **Logon Credentials** dialog box, where you can enter the account name and password of the account under which the host instance will run.
Disable host instance from starting	Select this check box to change the status of the selected host from enabled to disabled. Disabling a host instance is useful if you do not want the host instance to start, but you do want to preserve its settings.

After you install a host instance, you must start it so that it can route messages to the MessageBox databases. For information about starting a host instance, see How to Start a Host Instance.

How to Start a Host Instance

You can use the BizTalk Server 2006 Administration Console or Windows Management Instrumentation (WMI) to start host instances. After you add or stop a host instance, you must start it so that it is running and routing messages to the MessageBox databases.

For information about using WMI to start a host instance, see **MSBTS_HostInstance (WMI)**

Prerequisites

You must have the following user rights to create host instances, modify host instance properties, start a host instance, stop a host instance, and delete host instances:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Administrators group, see Managing the BizTalk Administrators Group .

- You must be a member of the Administrators group on the run-time computer.
- You must be a SQL Server administrator (a member of the sysadmin SQL Server role) on the computer running SQL Server.
- You must be a member of the db_accessadmin and db_securityadmin SQL Server database roles for the following databases:
  - BizTalk MessageBox (BizTalkMsgBoxDb) (all)
  - BizTalk Tracking (BizTalk DTADb)
  - Rule Engine (BizTalkRuleEngineDb)
  - BizTalk Management (BizTalkMgmtDb)
  - BAM Primary Import (BAMPrimaryImport)

To start a host instance

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Host Instances**.
3. In the details pane, right-click the host instance you want to start, and then click **Start**.

   The status of the host instance changes to **Start pending**. After the host instance initiates, the status changes to **Running**.

After you start a host instance, you can stop it to prevent it from routing messages to the MessageBox database. You must stop a host instance before you can remove BizTalk Server from a given computer. For information about stopping a host instance, see How to Stop a Host Instance

How to Stop a Host Instance

You can use the BizTalk Server 2006 Administration Console or Windows Management Instrumentation (WMI) to stop host instances. You must stop a host instance before you can delete it or remove BizTalk Server from a computer. You can stop a host instance that is installed and started.

For information about using WMI to stop a host instance, see **MSBTS_HostInstance (WMI)**

Prerequisites

You must have the following user rights to create host instances, modify host instance properties, start a host instance, stop a host instance, and delete host instances:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Administrators group, see Managing the BizTalk Administrators Group .
- You must be a member of the Administrators group on the run-time computer.
- You must be a SQL Server administrator (a member of the sysadmin SQL Server role) on the computer running SQL Server.
- You must be a member of the db_accessadmin and db_securityadmin SQL Server database roles for the following databases:
  - BizTalk MessageBox (BizTalkMsgBoxDb) (all)
  - BizTalk Tracking (BizTalk DTADb)
  - Rule Engine (BizTalkRuleEngineDb)
  - BizTalk Management (BizTalkMgmtDb)
  - BAM Primary Import (BAMPrimaryImport)

To stop a host instance

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Host Instances**.
3. In the details pane, right click the host instance you want to stop, and then click **Stop**.

   The status of the host instance changes to **Stopped**.

After you stop a host instance, you can start it, delete it, or remove BizTalk Server from the computer. For information about starting a host instance, see How to Start a Host Instance. For information about deleting a host instance, see How to Delete a Host Instance.

How to Delete a Host Instance

You can use the BizTalk Server 2006 Administration Console or Windows Management Instrumentation (WMI) to delete host instances.

For information about using WMI to delete a host instance, see **MSBTS_HostInstance (WMI)** .

When you delete a host instance, the instance of the BizTalk Server runtime is removed from the associated server and the BizTalk Management database is updated to remove that instance from the host.

To avoid losing messages when you delete a host instance, BizTalk Server completes all processing before the instance is actually removed.

Prerequisites

You must have the following user rights to create host instances, modify host instance properties, start a host instance, stop a host instance, and delete host instances:

- You must be a member of the BizTalk Server Administrators group. For information about adding users to the BizTalk Administrators group, see Managing the BizTalk Administrators Group .
- You must be a member of the Administrators group on the run-time computer.
- You must be a SQL Server administrator (a member of the sysadmin SQL Server role) on the computer running SQL Server.
- You must be a member of the db_accessadmin and db_securityadmin SQL Server database roles for the following databases:
  - BizTalk MessageBox (BizTalkMsgBoxDb) (all)
  - BizTalk Tracking (BizTalk DTADb)
  - Rule Engine (BizTalkRuleEngineDb)
  - BizTalk Management (BizTalkMgmtDb)
  - BAM Primary Import (BAMPrimaryImport)

To delete a host instance

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.
3. In the details pane, right click the host you want to delete, and then click **Delete**.
4. In the **Confirm delete host instance** dialog box, click **Yes**.

How to Modify Host Instance Properties

You can use the BizTalk Server 2006 Administration Console or Windows Management Instrumentation (WMI) to modify host instances. You can modify the service account running a host instance. You can also disable a host instance. For example, if you want to preserve the settings for a host instance and you do not want it to start, you can disable it.

Host instances of trusted hosts and host instances of non-trusted hosts cannot use the same service accounts. If you want to change the trust setting of a host instance and the host instance uses a service account that other host instances use, you can do one of the following:

- You can change the service account of the host instance for which you want to change the trust settings to a new service account.
- You can change the service account of the host instance to an existing service account that other host instances with the same trust setting use.
- You can delete the host instance, and re-create it with a different trust setting and service account.

For information about using WMI to modify a host instance, see **MSBTS_HostInstance (WMI)** .

Prerequisites

You must have the following user rights to modify host instance properties:

- You must be a member of the BizTalk Server Administrators group.
- You must be a member of the Administrators group on the run-time computer.
- You must be a SQL Server administrator (a member of the sysadmin SQL Server role) on the computer running SQL Server.
- You must be a member of the db_accessadmin and db_securityadmin SQL Server database roles for the following databases:
    - BizTalk MessageBox (BizTalkMsgBoxDb) (all)
    - BizTalk Tracking (BizTalk DTADb)
    - Rule Engine (BizTalkRuleEngineDb)
    - BizTalk Management (BizTalkMgmtDb)
    - BAM Primary Import (BAMPrimaryImport)

To modify host instance properties

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Host Instances**.

3. In the details pane, right click the host instance you want to modify, and then click **Properties**.

4. In the **Host Instance Properties** dialog box, click **Configure** to modify the service account information.

5. In the **Logon Credentials** dialog box, do the following, and then click **OK**:

Use this	To do this
Host name	Displays the name of the host associated with the selected server.
Server	Displays the server associated with the selected host.
Logon	Displays the account name of the new service account under which the host instance will run.
Configure	Click to display the **Logon Credentials** dialog box, where you can enter the account name and password of the account under which the host instance will run.
Disable host instance from starting	Select this check box to change the status of the selected host from enabled to disabled. Disabling a host instance is useful if you do not want the host instance to start, but you do want to preserve its settings.

Managing Servers

The Servers node in the BizTalk Server Administration Console lists all servers that belong to the BizTalk Server group. Using the procedures in this section, you can add or remove servers from the group, or move servers from one group to another.

In This Section

- How to Add a Server to a Group
- How to Move a Server from One Group to Another
- How to Remove a Server from a Group

How to Add a Server to a Group

You can use BizTalk Server Configuration to add a server to a BizTalk group. You add additional servers to a BizTalk group to scale out your BizTalk Server environment.

A server can only be associated with one BizTalk group. If a server already belongs to another group, you must first remove that server from its current group before you can add it to a new group. For information about removing a server from a BizTalk group, see How to Remove a Server from a Group.

Prerequisites

To perform this procedure, you must be logged on as a member of the BizTalk Server Administrators group and as a member of the Windows Administrators group.

To add a server to a BizTalk group

1. On the computer that you want to add to a BizTalk Server group to, click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Configuration**.

2. In the console tree, click **Group**.

3. In the details pane, select **Join an existing BizTalk Group**.

4. Continue the configuration of the server. Make sure that the databases to which you point the server are the databases for the BizTalk group the server is joining.

How to Move a Server from One Group to Another

server can only be associated with one BizTalk group. To move a server from one group to another, you must first remove the server from the original group, and then add it to the new one.

Prerequisites

To perform this procedure, you must be logged on as a member of the BizTalk Server Administrators group.

To move a server from one BizTalk group to another

1. On the computer that you want to move from the BizTalk group to another, open a command prompt. Click **Start**, click **Run**, type **cmd** and then click **OK**.

2. Navigate to the following directory:
%SystemRoot%\Program Files\Microsoft BizTalk Server 2006.

3. At the command prompt, type:

Configuration.exe /u

4. Press ENTER to remove the server from the current BizTalk group.

5. At the command prompt, type:

Configuration.exe

6. Press ENTER to start the Configuration Wizard.

On the **Welcome to BizTalk Server 2006 Configuration Wizard** page, click **Next**.

7. On the **Configuration Options** page, select **Join** to join an existing BizTalk group, select **No** in the **Is this the master secret server** drop-down list, and then click **Next**.

8. Continue the configuration of the server. Make sure that the databases to which you point the server are the databases for the BizTalk group the server is joining.

The server is now a member of the BizTalk group. You can now add host instances to this server

How to Remove a Server from a Group

A server can only be associated with one BizTalk group. If a server already belongs to another group, you must first remove that server from its current group before you can add it to a new group.

Prerequisites

To remove a server from a group

1. On the computer that you want to remove from the BizTalk group, open a command prompt. Click **Start**, click **Run**, type **cmd** and then click **OK**.

2. Navigate to the following directory:
%SystemRoot%\Program Files\Microsoft BizTalk Server 2006.

3. At the command prompt, type:

Configuration.exe /u

4. Press ENTER to remove the server from the current BizTalk group.

The Configuration Wizard appears.

5. On the **Welcome to BizTalk Server 2006 Configuration Wizard** page, click **Finish**.

Managing MessageBox Databases

The MessageBox database has three essential functions. It stores subscriptions and tracking information and it delivers the messages to the services that match the subscriptions. The MessageBox database is a host platform that stores the queues and state tables for each BizTalk Host. The MessageBox database also stores messages and message properties.

If the MessageBox databases are an asset with high-risk exposure in your environment, we recommend that you use Internet Protocol security (IPSec) or Secure Sockets Layer (SSL) to restrict and secure communication to and from the MessageBox databases.

If you use Windows Server 2003, you must enable distributed transaction coordinator (DTC) on the MessageBox database. You must also enable network clients for multi-computer deployments. For more information, see **Ports for the Administration Server** .

This section contains procedural information about managing MessageBox databases in your environment. For conceptual information about MessageBox databases and the subscription model Microsoft BizTalk Server 2006 uses to process messages, see The MessageBox Database.

In This Section

- How to Add a New MessageBox Database
- How to Disable New Message Publication
- How to Delete a MessageBox Database

How to Add a New MessageBox Database

You can use the BizTalk Server 2006 Administration Console to add a new MessageBox database to your BizTalk Server deployment. MessageBox databases are the basis for load-balancing work items across servers that do cooperative processing. To increase the number of messages that your system can process, you may need to add additional MessageBox databases.

You cannot create a new MessageBox database and enlist orchestrations, send ports, or send port groups at the same time. Enlisting orchestrations, send ports, or send port groups accesses data that BizTalk Server must copy to the new MessageBox database. While this data is being accessed, BizTalk Server cannot copy it into the new MessageBox database.

You can designate both local and remote databases as MessageBox databases. For information about BizTalk Server databases, see Databases in BizTalk Server.

Prerequisites

Administrators who manage MessageBox databases must have the required user rights. You must have the following user rights to manage MessageBox databases and disable new message publication:

- You must be logged on as a member of the BizTalk Server Administrators group.
- You must be a SQL Server Administrator on the computer where the database exists.

To add a new MessageBox database

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, and then click **Platform Settings**.
3. Right-click **Message Boxes**, click **New**, and then click **Message Box**.
4. In the **Message Box Properties** dialog box, do the following, and then click **OK**:

Use this	To do this
SQL Server	Displays the name of the SQL server that hosts the MessageBox database.
Database	Displays the name of the MessageBox database.
Master subscription message box	Indicates whether the selected MessageBox database is the master. If the current MessageBox database is the master, this check box is selected and unavailable. The first MessageBox database created when you run the Configuration Wizard is the master by default.
Disable new message publication	Select this check box to specify that you do not want this MessageBox database to receive activation messages.

How to Disable New Message Publication

You can use the BizTalk Server 2006 Administration Console or Windows Management Instrumentation (WMI) to disable new message publication. You disable new message publication in the MessageBox database to stop the receipt of new messages by the MessageBox database. In some BizTalk Server environments, you can improve performance if you disable new message publication for the master MessageBox database. Disabling new message publication does not affect existing messages in the MessageBox database or service instances that are in progress.

For information about using WMI to disable new message publication, see **MSBTS_MsgBoxSetting.DisableNewMessagePublication Property (WMI)**.

Prerequisites

Administrators who manage MessageBox databases must have the required user rights. You must have the following user rights to manage MessageBox databases and disable new message publication:

- You must be logged on as a member of the BizTalk Server Administrators group.
- You must be a SQL Server Administrator on the computer where the database exists.

To disable new message publication

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Message Boxes**.
3. In the details pane, right-click the MessageBox database you want to stop, and then click **Properties**
4. In the **Message Box Properties** dialog box, select the **Disable new message publication** check box, and then click **OK**.

How to Delete a MessageBox Database

You use the BizTalk Administration console or Windows Management Instrumentation (WMI) to remove a MessageBox database from a BizTalk group. You can remove a MessageBox database from a BizTalk group, or you can delete it from your BizTalk Server deployment entirely.

For example, you can delete a MessageBox database you are no longer using, such as a database used for testing purposes.

There are eight steps to permanently and completely removing MessageBox databases from your BizTalk Server deployment:

1. Disable new message publication

   You must disable the publication of new messages before you delete a MessageBox database. For information about disabling new message publication, see How to Disable New Message Publication.

2. Wait for the cache refresh interval to expire

   After you disable the publication of new messages, you must wait before you delete the database. The wait time is defined as twice the length of the CacheRefreshInterval (2*CacheRefreshInterval). The default value of CacheRefreshInterval is 60 seconds. You use the **Microsoft BizTalk Server 2006 Properties** dialog box to change the CacheRefreshInterval property.

3. Remove the MessageBox database from the BizTalk Group.

   Removing the MessageBox database from the BizTalk Group removes the MessageBox reference from the BizTalk Management database.

4. Restart host instances that contain cached connections to the MessageBox database.

   You must restart the host instance before physically deleting the database from SQL Server if cached database connections from the run-time engine are present. For information about starting a host instance, see How to Start a Host Instance.

5. Stop all in-progress host instances that access the database.

   If you are removing a non-primary MessageBox database, you must first remove the MessageBox database reference from the BizTalk Server management databases, and then stop all in-progress host instances before removing the database from SQL Server. For information about stopping an in-progress host instance, see How to Stop a Host Instance.

6. Ensure that the background SQL Server Agent job is finished.

   Before you permanently delete a MessageBox database from your BizTalk Server deployment, you should first ensure that the background SQL Server Agent job has finished transferring all tracked message bodies to the TrackingSpool table, and then back up the TrackingSpool tables. For information about checking the status of a background SQL Server Agent job, see SQL Server Books Online.

7. Back up the TrackingSpools tables.

   Tracked message bodies remain in the MessageBox database until you manually back up the TrackingSpool tables into external storage. Before the backup happens, a background SQL Server Agent job transfers the message bodies from the Spool table to the TrackingSpool table. For information about manually backing up SQL Server tables, see SQL Server Books Online.

8. Remove the database from SQL Server.

   Deleting a MessageBox database from a BizTalk Group does not physically remove the database from Microsoft SQL Server. To permanently delete the MessageBox database, you must remove it by using SQL Server Enterprise Manager after it is removed from the BizTalk Group.

Prerequisites

Administrators who manage MessageBox databases must have the required user rights. You must have the following user rights to manage MessageBox databases and disable new message publication:

- You must be logged on as a member of the BizTalk Server Administrators group.
- You must be a SQL Server Administrator on the computer where the database exists.

To delete a MessageBox database from a BizTalk Group

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Message Boxes**.

3. In the details pane, right-click the message box database you want to delete, and then click **Properties**

4. In the **Message Box Properties** dialog box, select the **Disable new message publication** check box.

5. Use the Health and Activity Tracking (HAT) tool to verify that no message instances are dehydrated or suspended on the MessageBox database you are deleting. For information about using HAT tool, see Health and Activity Tracking.

6. Wait for a period of time twice the length of the CacheRefreshInterval (2*CacheRefreshInterval). The default value of CacheRefreshInterval is 60 seconds.

7. In the details pane, right-click the MessageBox database that you want to delete, and click **Delete**.

8. After reading the warning message, click **OK**.

To delete a MessageBox database from SQL Server

1. Open the **BizTalk Administration Console**. Click **Start**, click **Run**, type **btsmmc.msc**, and then click **OK**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.

3. In the details pane, right-click all running host instances, and stop and restart each one.

4. On the server where the message box database resides, open SQL Server Enterprise Manager. Click **Start**, click **Run**, type **"SQL Server Enterprise Manager.msc"** and then click **OK**.

5. Open the appropriate server by clicking it, and then click **Databases**.

6. In the details pane, right-click the message box database you want to delete and click **Delete**.

Using Adapters

This section contains comprehensive information about adapters. It describes how adapters are used in Microsoft BizTalk Server 2006 and how to configure adapter handlers, send ports,

and receive locations for each adapter. It provides batching support information to help you understand and use the native adapters in your BizTalk solution.

In This Section

- Adapters in BizTalk Server
- Security Considerations for Adapters
- Base EDI Adapter
- BizTalk Message Queuing (MSMQT) Adapter
- File Adapter
- FTP Adapter
- HTTP Adapter
- MQSeries Adapter
- MSMQ Adapter
- POP3 Adapter
- SMTP Adapter
- SOAP Adapter
- SQL Adapter
- Windows SharePoint Services Adapter
- Creating and Deleting Adapter Handlers
- Using the BizTalk Adapter Trace Utility

Adapters in BizTalk Server

One of the primary design goals of BizTalk Server 2006 is to facilitate the exchange of business documents between trading partners. To help meet this goal, BizTalk Server 2006 includes several adapters that provide connectivity between BizTalk Server and trading partners using commonly recognized data protocols and document formats. This topic discusses what an adapter is and why you use an adapter.

What Is an Adapter?

An adapter is a software component that enables you to easily send messages out of or receive messages into BizTalk Server with a delivery mechanism that conforms to a

commonly recognized standard, such as SMTP, POP3, FTP, or Microsoft Message Queuing (MSMQ). As Microsoft BizTalk Server 2006 has evolved, the need for adapters that quickly enable connectivity with commonly used applications and technologies has increased.

BizTalk Server 2006 includes the following adapters, which are referred to as the "native" or "integrated" adapters: Base EDI, BizTalk Message Queuing, FILE, FTP, HTTP, MQSeries, MSMQ, POP3, SMTP, SOAP, SQL, and Windows Sharepoint Services. Native adapters are installed with BizTalk Server 2006. You can also create custom adapters for your specific solutions by using the BizTalk Adapter Framework.

Each of the native adapters is associated with a receive location designed to listen for messages from a certain transport at a certain address. After the message is received by the receive location, it is passed to the adapter. The adapter attaches the data stream to the message (typically in the body part of the message), adds any metadata pertaining to the endpoint that the data was received from, and then submits that message into the BizTalk Messaging Engine.

By default, when you run the BizTalk Configuration Wizard, the wizard installs the native adapters and creates an adapter handler with a default configuration for each one.

Using BizTalk Explorer and the BizTalk Server Administration console, you can modify the default configuration for the adapter handlers as well as add, remove, and modify send ports and receive locations for the adapters. For more information about these processes, see the appropriate topics in See Also.

Why Use an Adapter?

Using adapters greatly simplifies the transfer of messages into or out of BizTalk Server. If your existing infrastructure uses any of the transports for which there is a corresponding BizTalk adapter, then the process of sending messages to or receiving messages from BizTalk Server is as simple as configuring the appropriate adapter to send or receive messages with the corresponding transport mechanism.

Summary of Functionality Supported by BizTalk Adapters

The following table lists the primary benefit of each native adapter and whether the adapter provides the following features:

- **Transaction support.** The ability to send and receive documents under the context of a Microsoft Distributed Transaction Coordinator (MSDTC) transaction. This functionality is required for maintaining ordered message delivery and to guarantee that documents are not duplicated or lost.

- **Two-way communication support (Request/Response or Solicit/Response).** The ability to send a document and process a response message from the destination or to receive a document and send a response message to the source.

- **In-order receive support.** The ability to publish received documents to the BizTalk MessageBox database in the exact order that the documents were received.

- **SSO enabled.** The ability to use SSO authentication when sending or receiving documents with the adapter.

Adapter	Primary benefit	Transaction support	Two-way communication support	In-order receive support	SSO enabled
File	Easy to use.	No	No	No	No
FTP	Is widely used for business-to-business communications.	No	No	No	Yes
HTTP(s)	Is widely used for business-to-business communications.	No	Request/Response and Solicit/Response	No	Yes
SOAP	Supports the use of Web services.	No	Request/Response and Solicit/Response	No	Yes
MSMQT	Supports guaranteed once-only delivery of messages between BizTalk Server and Microsoft Message Queuing.	Yes	No	No	No
MSMQ	Supports guaranteed once-only delivery of messages between BizTalk Server and Microsoft Message Queuing.	Yes	No	Yes	No
MQ Series	Supports guaranteed once-only delivery of messages between BizTalk Server and IBM WebSphere MQ for Windows platforms.	Yes	No	Yes	Yes
SQL	Supports direct communication between BizTalk Server and SQL Server databases.	Yes	Solicit/Response only	No	No
Windows SharePoint Services	Supports the use of Windows SharePoint Services.	No	No	No	No

POP3	Supports receiving documents through e-mail.	No	No	No	No
SMTP	Supports sending documents through e-mail.	No	No	No	No
EDI	Supports processing of business documents that conform to the EDI standard.	No	No	No	No
Custom	Supports your system.	Yes, requires custom code.	Yes, requires custom code.	Yes, requires custom code.	Yes, requires custom code.

Security Considerations for Adapters

This section contains information about creating and maintaining adapter security.

In This Section

- Best Practices for Securing Adapters
- SSO for Native Adapters

Best Practices for Securing Adapters

This topic provides a list of best practices for adapter security.

Do not install untrusted adapters on your computer; use only certified adapter development partners.

For a list of available adapters from adapter development partners, see http://go.microsoft.com/fwlink/?LinkId=15279.

Do not store sensitive customer data in the default adapter schema.

You should configure the user name and password information only after you deploy an adapter. This ensures that the information gets stored in the Credential database. For more information about the Credential database, see Using SSO .

Grant the following permissions on the shared folders (the Receive folder and Send folder) that are used to pick up and drop files using the native adapters:

- Receive folder

  The service account for the BizTalk Host that picks up the file should have the following permissions at the file-system level:

  - List Folder / Read Data
  - Delete SubFolder and Files

  If the receive folder is on a network share, the following permissions must be granted at the file-share level:

  - The service account for the BizTalk Host that picks up the file must have Full Control permissions.
  - BizTalk Server administrators must have Full Control permissions for troubleshooting.
  - The external user or programs that drop files to this location must have Write permissions.

- **Send folder**

  - The service account for the BizTalk Host or Hosts that drop files here must have Write permissions.
  - BizTalk Server administrators must have Full Control permissions.
  - The external user or program that picks up files must have Read permissions.

Ensure that custom adapters display only nonsensitive data in property pages.

When the BizTalk Server Administration console or BizTalk Explorer asks for the adapter information for the File, HTTP, SOAP, SMTP, or BizTalk Message Queuing adapters, the messaging engine masks the sensitive information as VT_NULL before the messaging engine passes it to the object model. For adapters built using the BizTalk Adapter Framework, such as SQL, FTP, Base EDI, or custom adapters, the functionality is different. All information goes back to the adapter as a single unit of binary large objects (BLOB). The developer for the custom adapter is responsible for parsing this information and displaying only the nonsensitive data in the property pages.

Add the user account under which the EDI service is running to the BTS_HOST_USERS SQL role.

This is required so that you can obtain BizTalk Explorer Object Management (OM) Access without administrative permissions. To do this, add "EDI Subsystem Users" to the BTS_HOST_USERS role in the BizTalk Management database, BizTalkMgmtDb. To add "EDI Subsystem Users" to the BTS_HOST_USERS role, complete the following steps:

- Launch the SQL Enterprise Manager from **Start**, **Programs**, **Microsoft SQL Server**, **Enterprise Manager**.
- Connect to the SQL Server that houses your BizTalk Management database.
- Expand this server in the Enterprise Manager.
- Expand **Databases**, and then expand the BizTalk Management database.
- Click **Roles**.
- In the details pane, right-click the BTS_HOST_USERS role, and then click **Properties**.
- Click **Add**, and then click the EDI Subsystem Users group to add it.

If the EDI Subsystem Users group is not available in the list of users to add, you must add the EDI Subsystem Users group as a new database user to the BizTalk Management database. To add the EDI Subsystem Users group as a new database user, complete the following steps in the SQL Enterprise Manager:

- Expand the BizTalk Management database.
- Right-click **Users** and click **New Database User**.
- In the drop-down box for **Login name**, select the **EDI Subsystem Users** group, and then click **OK**.

Add the user account under which the BizTalk service is running to the EDI Subsystem Users group

SSO for Native Adapters

Enterprise Single Sign-On (SSO) enables you to sign on only once when interoperating with different computer systems or Web sites. This feature of Microsoft BizTalk Server 2006 enables BizTalk adapters to provide the appropriate user ID and credentials to multiple applications within your network that use a common authentication mechanism based on your Microsoft Windows credentials. After Windows authenticates your credentials, you do not need to provide additional credentials to connect to the applications.

SSO is available for the HTTP and SOAP adapters, although it is disabled by default. For the HTTP adapter, you can configure the send port and receive location to use SSO; for the SOAP adapter, you can configure only the receive location to use SSO. For both adapters, you use the BizTalk Server Administration console or BizTalk Explorer to configure SSO.

Authentication in SSO relies primarily on Windows authentication and the Windows groups created in Active Directory. All operations completed by a user or administrator with SSO require that Windows authenticate the user or administrator first.

For more information about how SSO works with the HTTP and SOAP adapters, see the appropriate topics in See Also.

Base EDI Adapter

The Microsoft BizTalk Server 2006 Base EDI adapter provides comprehensive electronic data interchange (EDI) messaging functionality to complement the XML messaging capabilities of BizTalk Server 2006.

You can use the Base EDI adapter to:

- Send and receive documents in an EDI format.
- Create partner-specific XML schemas.
- Track the processing history of an individual document.
- Automatically acknowledge the receipt of an EDI message.

In This Section

- What Is the Base EDI Adapter?
- Verifying the Syntax of an EDI Message
- Translating an EDI Message
- Base EDI Adapter Tutorials
- Base EDI Adapter Operations
- Configuring the Base EDI Adapter
- Developing EDI Business Processes
- Developing Schemas and Maps for the Base EDI Adapter
- Using EDI Acknowledgements
- Working with Custom Reference Numbers
- Using EDIFACT EDI Segments
- Working with X-12 EDI Documents
- Working with EDI Annotations
- Base EDI Adapter Supported Standards

- Base EDI Adapter Security Recommendations
- Troubleshooting the Base EDI Adapter
- Creating Custom EDI Schemas

What Is the Base EDI Adapter?

The Base EDI adapter consists of the send and receive handlers and the EDI subsystem. This adapter is unique among the BizTalk Server 2006 integrated adapters in that it is the only integrated adapter that is not designed to deliver documents to or from a system by using a particular delivery mechanism. All of the other BizTalk Server integrated adapters facilitate document transfer between BizTalk Server and other systems by using a delivery mechanism such as e-mail, the Windows file system, or Message Queuing. The Base EDI adapter is designed to facilitate only the processing of documents that use the EDI format.

What the Base EDI Adapter Provides

Many companies have made substantial investments in EDI over a period of years. Maintaining an EDI solution, however, can create problems such as:

- Value-added networks (VANs) are expensive to use for communication, but there is no viable alternative.
- Providers charge high maintenance fees.
- Existing EDI suppliers may stop supporting the software due to mergers, takeovers, closures, and so on.

The BizTalk Server 2006 Base EDI adapter has been developed to address these problems by providing the following:

- The ability to migrate from VANs to more cost-effective solutions such as AS2, which enables you to send encrypted data to your trading partner using HTTP
- Full support from Microsoft
- Full auditing and tracking in one place for all messaging and monitoring
- Interoperability with an unlimited number of trading partners
- Comprehensive repository of EDI document standards
- Automated parsing and serializing between EDI and XML

Supported EDI Standards

The Base EDI adapter supports the two most widely implemented EDI standards: X-12 and EDIFACT.

X-12 Standard

The X-12 standard is a messaging standard developed by the American National Standards Institute (ANSI). ANSI X-12 is mainly used within the United States. An ANSI X-12 document is structured, so that it must always contain the following segments:

- ISA segment
- GS segment
- ST segment
- SE segment
- GE segment
- IEA segment

The ANSI releases several new editions of the X-12 standard each year.

EDIFACT Standard

EDIFACT stands for Electronic Data Interchange for Administrative Commerce and Transport. EDIFACT is a world-wide EDI standard that has been defined by the United Nations. EDIFACT, or its derivatives, is the mostly widely implemented EDI standard in the world.

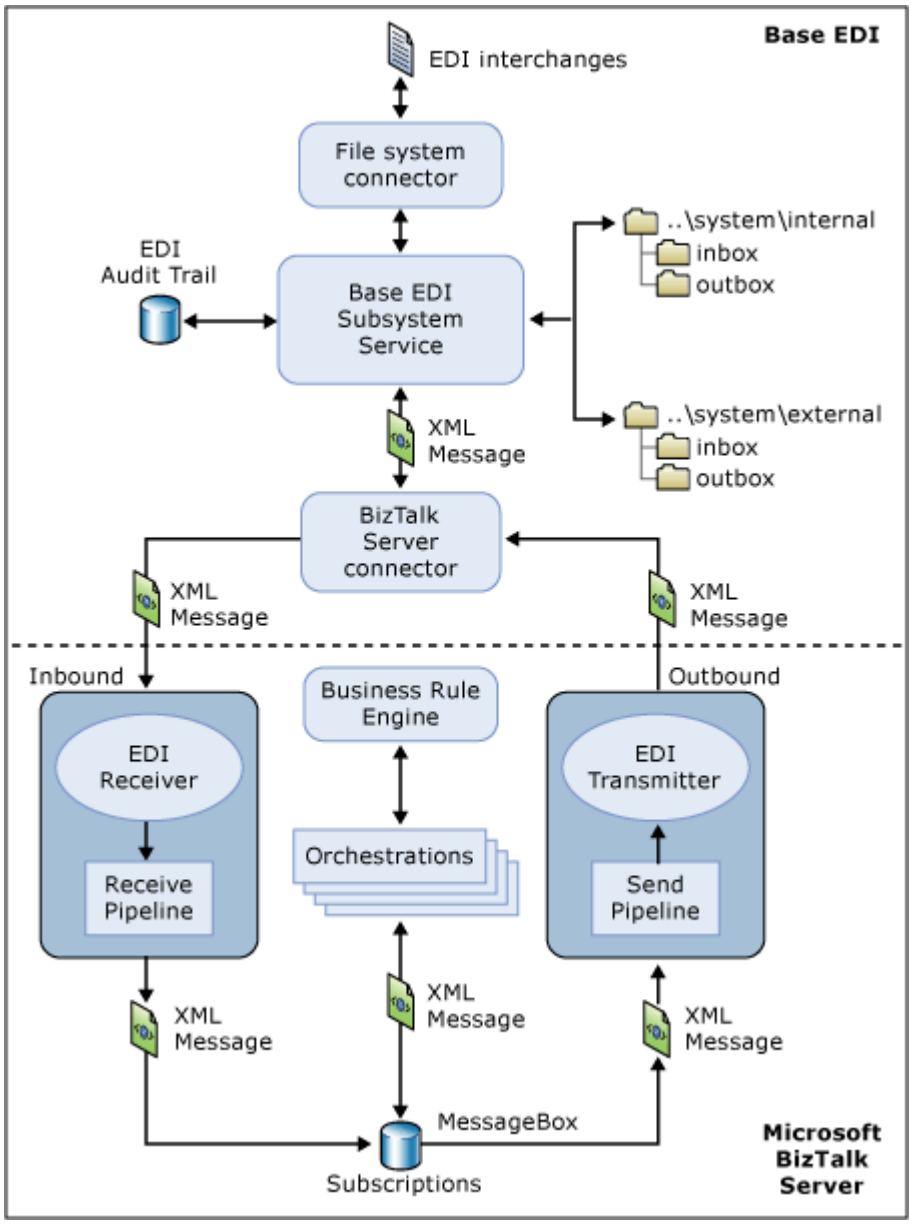
An EDIFACT document is structured, so that it must always contain the following segments:

- UNA segment
- UNB segment
- UNG segment
- UNH segment
- UNT segment
- UNE segment

The United Nations releases several new editions of the EDIFACT standard each year. For information about EDIFACT standards supported by BizTalk Server 2006, see Supported EDIFACT Standards .

How the Base EDI Adapter Works with BizTalk Server

The following diagram illustrates the relationship between BizTalk Server and the Base EDI adapter.



Document Flow Through BizTalk Server

This section provides examples of the flow of a document through BizTalk Server and the EDI subsystem using three different scenarios: Inbound EDI to Outbound EDI, Inbound XML to Outbound EDI, and Inbound EDI to Outbound XML.

Inbound EDI to Outbound EDI

1. An EDI document is dropped into the folder monitored by the Base EDI adapter. The default location for this folder is C:\Documents and Settings\All Users\Application Data\Microsoft\BizTalk Server 2006\EDI\Subsystem\Documents\PickupEDI\. You can configure this location in the EDI adapter receive handler connector properties.

2. The EDI subsystem picks up the document and parses it to determine the sender and receiver and to verify the syntax of the document type:

   - For an X12 document, the sender and receiver URI values are read from the ISA elements of the inbound document. The sender is defined in the ISA05, ISA06, and GS02 elements. The receiver is defined in the ISA07, ISA08, and GS03 elements.

   - These values are compared to the list of EDI URI values for the parties that have been created in the BizTalk Server Administration console.

   - After the sender, receiver, and document type are verified, the EDI subsystem processes the document and translates it to XML. Then the EDI receive handler persists the document to the BizTalk MessageBox database. If no match is found for the sender and receiver, or if the document syntax is invalid, the EDI subsystem suspends the document.

3. The BizTalk send ports that are associated with the receiving party subscribe to the document and pick up the document for outbound processing.

4. A send port that processes the document uses a **Transport Type** of **EDI**, which is how the document is routed back to the EDI subsystem for processing.

5. After the EDI subsystem processes the document, it sends the document to the location specified in the file system parameters for the send port.

Inbound XML to Outbound EDI

1. An XML document is dropped into a location that is monitored by a receive location.

2. The document is picked up by the receive location. The receive location is tied to a receive port.

3. The document is persisted to the BizTalk MessageBox database and the subscribing send port picks up the document.

4. The send port that processes the document uses a **Transport Type** of **EDI**, which is how the document is routed to the EDI subsystem for processing.

5. After the EDI subsystem processes the document, it sends the document to the location specified in the file system parameters for the send port.

Inbound EDI to Outbound XML

1. An EDI document is dropped into the folder monitored by the Base EDI adapter. The default location for this folder is C:\Documents and Settings\All Users\Application Data\Microsoft\BizTalk Server 2006\EDI\Subsystem\Documents\PickupEDI\. You can configure this location in the EDI adapter receive handler connector properties.

2. The EDI subsystem picks up the document and parses it to determine the sender and receiver and to verify the syntax of the document type:

- For an X12 document, the sender and receiver URI values are read from the ISA elements of the inbound document. The sender is defined in the ISA05, ISA06, and GS02 elements. The receiver is defined in the ISA07, ISA08, and GS03 elements.

- These values are compared to the list of EDI URI values for the parties that have been created in the BizTalk Server Administration console.

- After the sender, receiver, and document type are verified, the EDI subsystem processes the document and translates it to XML. Then the EDI receive handler persists the document to the BizTalk MessageBox database. If no match is found for the sender and receiver, or if the document syntax is invalid, then the EDI subsystem suspends the document.

3. The BizTalk send ports that are associated with the receiving party subscribe to the document and pick it up for outbound processing.

4. A BizTalk send port processes the document and hands it off to the transport that it is configured to use. For outbound XML the send port uses the XMLTransmit pipeline.

Verifying the Syntax of an EDI Message

Syntax validation is the first step in the process of converting a document from EDI to XML or vice versa. After the syntax of the document has been validated, the translation process begins.

To ensure that documents are processed correctly, the BizTalk Server 2006 Base EDI adapter checks that the syntax of each document matches the syntax description for that document in the document definitions repository of the Base EDI adapter. The following paragraphs describe the syntax verification criteria that the Base EDI adapter uses.

Illegal Characters

Each document is verified to ensure that all the characters in the document are allowed according to the character set defined for the document format.

For EDIFACT documents of syntax version UNOA, characters A-Z, 0-9, blank and . , ( ) / - = are allowed.

For syntax version UNOB, characters a-z, A-Z, 0-9, blank and . , ( ) / - = : + ` ? are allowed.

All other EDIFACT syntaxes are linked to the standard ISO character sets.

For ANSI X-12 documents, characters A-Z, 0-9, blank and !"&'()*+,-./:;?= are allowed and, if desired, a-z, % ~ @ [ ] _ { } \ | < > # $.

Furthermore, the following nonprintable characters are allowed within X-12 (hexadecimal notation): 07, 09, 0A, 0B, 0C, 0D, 1C through 1F, and if desired, 01 through 06 and 11 through 17. The other (proprietary) formats can also be linked to character sets.

Separator Usage

The locations and the correct usage of the separators defined for the document format are checked.

Numeric Elements

Only one character for the decimal notation is allowed in a numeric element, and the decimal notation must be coherent with the document definition. A sign (-) is only allowed for signed elements. The position of the sign must be coherent with the document type or format version definition (begin or end).

Elements of Target Document

For the target document, the Base EDI adapter verifies that all mandatory singular elements are present, that all mandatory composite elements are present, and that all mandatory components are present when a conditional composite element is present.

Elements of Source Document

For the source document, the Base EDI adapter verifies that all mandatory singular elements are present, that all mandatory composite elements are present, and that all mandatory components are present when a conditional composite element is present.

Segments

Documents are checked to verify that all mandatory segments are present. The Base EDI adapter verifies that the number of occurrences for each segment is correct. Occurrence 0 (zero) means infinite.

Qualifiers and Code Sets

Segments are checked to verify that all mandatory qualifiers and codes are present and that the qualifiers and codes read are allowed according to the specified set for each element.

Number of Documents and Segments

For EDIFACT and X-12 interchanges, the interchange control count in the UNZ/ IEA segment must be equal to the number of documents in the interchange. For documents within an interchange, the number of segments in the UNT/ SE segment must be equal to the number of segments in the document. The same rules apply to the functional group headers and trailers.

References

For EDIFACT and ANSI X-12 interchanges, the interchange control reference in the UNB/ ISA segment must be equal to the interchange control reference in the UNZ/ IEA segment. For documents within an interchange, the document reference number in the UNH/ ST segment

must be equal to the document reference number in the UNT/ SE segment trailer. The same rules apply to the functional group headers and trailers.

Headers and Trailers for EDIFACT

The UNA segment does not have to be present. The first segment or the first segment after the UNA segment must be the UNB segment. Each document has to start with the UNH segment and has to end with the UNT segment. The interchange has to end with the UNZ segment. Functional groups are allowed (starting with UNG and ending with UNE).

Headers and Trailers for ANSI X-12

The first segment of an interchange must be the ISA segment. Each group of documents should be enclosed between the GS and GE segments. Each document should start with the ST segment and end with the SE segment. The interchanges should end with the IEA segment.

Translating an EDI Message

At this point, the document has been identified and the syntax of the document has been verified as correct. This section describes how the BizTalk Server 2006 Base EDI adapter processes the values of an EDI document and converts the values to or from the XML format.

After the translation process is completed, a functional acknowledgment is sent to the sender of the transaction set and the document can be sent to a trading partner.

Element Type Conversion

The general rule is: As long as it is possible at run time, the Base EDI adapter translates any element type to any element type.

Numeric or Signed to Numeric or Signed

The length and number of decimals may vary, as long as it is possible at run time.

Date to Date

Dates can be in the form ddmmyy, mmddyy, yymmdd, ddmmyyy, mmddyyyy, or yyyymmdd.

All translation combinations are possible. If a date element without a century has been mapped to a date element with a century, the following rule applies: If the year is < 50, the century will be 2000, else the century will be 1900. If a date element with a century has been mapped to a date element without a century, the century will be stripped.

Time to Time

Times can be in the form hhmmss or hhmm. All translation combinations are possible.

Booleans

Booleans can be either numeric (BN) or alphanumeric (BA). Booleans can be converted to each other, and can also be converted to and from alphanumeric and numeric elements, as long as it is possible at run time.

Translation of Levels and Segments

Elements from more than one segment can be translated to one segment and elements from one segment can be translated to several segments. Elements of a segment can be translated to elements of a segment on any another nesting level.

Base EDI Adapter Tutorials

These tutorials contain detailed information about how the Microsoft BizTalk Server 2006 Base EDI adapter can be used within your company to facilitate enterprise application integration (EAI), and how it can be used among business partners to automate business-to-business procurement solutions. Tutorial 1 covers an end-to-end document translation, including sending a functional acknowledgment using a predefined EDI solution. Tutorial 2 builds on those concepts, showing you how to create an end-to-end document translation without an orchestration.

Tutorial 1: EDI-to-XML Document Translation

The lessons in this tutorial discuss new concepts, terminology, and the tools and services that are used to develop an EDI solution using the Base EDI adapter.

This tutorial also covers the processes and tasks that are required to build an EDI solution. This gives you the opportunity, through hands-on experience, to see how an EDI solution is constructed.

This tutorial covers:

- The basics of an EDI solution
- Developing and testing the EDI solution

To begin working on this tutorial, go to Tutorial 1: EDI-to-XML Document Translation .

Tutorial 2: XML-to-EDI Document Translation

After you are familiar with the basics of the concepts and tools, proceed to the lessons in Tutorial 2 to learn how to use BizTalk Server 2006 to create an XML-to-EDI scenario. These lessons walk you through the configuration steps and cover more advanced features than the first tutorial, such as sending an acknowledgment for the EDI document.

To begin working on this tutorial, go to Tutorial 2: XML-to-EDI Document Translation .

In This Section

- Tutorial 1: EDI-to-XML Document Translation
- Tutorial 2: XML-to-EDI Document Translation

Tutorial 1: EDI-to-XML Document Translation

This tutorial covers an EDI-to-XML document interchange. The test document is picked up by the Microsoft BizTalk Server 2006 Base EDI adapter in an EDI document format and then translated into an XML document.

Before you begin these lessons, you should be familiar with the fundamental concepts of the Base EDI adapter and with the tools and processes that you need to start building solutions with BizTalk Server 2006.

Review the following documentation before you begin these lessons:

- Base EDI Adapter Operations
- Base EDI Adapter

Requirements

- To successfully complete this tutorial, the Base EDI adapter and all its components must be installed on your computer. For more information, see Installing BizTalk Server 2006.
- Ensure that the BizTalk Base EDI service has been started. To do so, click **Start**, point to **Settings**, and then click **Control Panel**. In Control Panel, double-click **Administrative Tools**. In Administrative Tools, double-click **Services**.
- The necessary tutorial files can be found at <*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI.
- Ensure that you are part of the EDI subsystem user group before beginning this tutorial. Contact your administrator for details.

User Account Prerequisites

When using the Configuration Wizard, you must configure the BizTalk Base EDI service using the same logon credentials as the BizTalk Server Host service.

To ensure that the user account has sufficient permissions to access the EDI and BizTalk Management databases, the user who configures the EDI send handler must be part of the appropriate NT Users groups, namely the EDI Subsystem and BizTalk Server Administration user groups.

To access the BizTalk Explorer Object Model as a non-administrator, do the following:

1. Add the EDI Subsystem Users group to the BTS_HOST_USERS role found in the BizTalkMgmtDb BizTalk Management database.

2. When developing EDI solutions ensure that:

   - The logged-on user is a part of the EDI users group.
   - If the EDI service is running as a local machine account, you must log on and develop using a local machine account.

In This Section

- Lesson 1: Plan the EDI-to-XML Solution
- Lesson 2: Configure the EDI-to-XML Solution
- Lesson 3: Run the EDI-to-XML Solution

Lesson 1: Plan the EDI-to-XML Solution

BizTalk Server 2006 provides a development and execution environment for electronic data interchange (EDI), both within and between businesses.

EDI Scenario

This tutorial module outlines an EDI solution for two trading partners who have different methods of formatting documents electronically and who need to electronically exchange data. The parties central to this interchange are Northwind and Contoso.

Northwind Airlines recently announced that it would be sending schedule and fare information for all of its flights in an XML document. Contoso, a trading partner of Northwind, can only accept electronic documents that are formatted according to the X-12 EDI document standard.

The difficulty here is that the companies have adopted standards that are not mutually compatible. The BizTalk Server 2006 Base EDI adapter acts as a translator for interchanges between the two companies.

In this module, Contoso initiates an interchange by sending a document in the X-12 document format to Northwind. The Base EDI adapter receives the EDI document and translates it into an XML representation of itself. The XML document is then sent to Northwind.

To begin configuring the document interchange, go to Lesson 2: Configure the EDI-to-XML Solution .

Lesson 2: Configure the EDI-to-XML Solution

In this lesson, you will create an electronic data interchange (EDI) solution that enables two trading partners to exchange data, even though the parties use different formats for the transfer of electronic data.

To begin configuring the EDI-to-XML solution, go to next topic Step 1: Start the BizTalk Base EDI Service .

In This Section

Step 1: Start the BizTalk Base EDI Service

By default, the BizTalk Base EDI service is set to Disabled. You must start this service to run the EDI solution.

To start the BizTalk Base EDI service

1. Click **Start**, point to **Settings**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**.
3. In Administrative Tools, double-click **Services**.
4. Right-click **BizTalk Base EDI service**, and then click **Properties**.
5. In the **Startup type** dialog box, select **Manual** from the drop-down list box.

6. Right-click **BizTalk Base EDI service**, and then click **Start**.

Step 2: Open the EDI Solution

The EDI solution has certain predefined aspects, such as a preconfigured orchestration, mapping, and document definitions. You must open this solution to begin.

To open the EDI solution in Visual Studio

1. Click **Start**, point to **Programs**, and then click **Microsoft Visual Studio 2005**.
2. On the **File** menu, click **Open Solution**.
3. Navigate to **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Visual Studio Projects\Getting Started with EDI**.
4. Double-click **Getting started with EDI.sln**.
5. On the **View** menu, click **BizTalk Explorer**.

Step 3: Create Trading Parties

The first configuration step is to create the parties that will be exchanging documents, namely Contoso and Northwind. In this tutorial, Contoso initiates the document interchange by sending an EDI message to Northwind.

To add a new party called Contoso

1. Right-click the **Parties** folder, and then click **Add party**.
2. In the **Name** field at the top of the dialog box, type **Contoso**.

   In this scenario, Contoso is the sender of the EDI documents.
3. For the **Party Alias** values, type **EDI** in the name field by clicking the row with the * symbol.
4. In the **Qualifier** field, type **EDI**.

   For parties that will exchange EDI documents, you must use an address that begins with "EDI://". BizTalk Server reserves this prefix for parties that exchange EDI documents using the Base EDI adapter. The qualifier/value pair defined for each party must be unique.
5. In the **Value** field, type **EDI://7654321:ZZ:7654321**.
6. Click **OK**.

To add a new party called Northwind

1. Right-click the **Parties** folder, and then click **Add party**.
2. In the **Name** field at the top of the dialog box, type **Northwind**.

   In this scenario, Northwind is the home organization (recipient) of the XML documents that are converted from native EDI format.
3. For the **Party Alias** values, type **EDI** in the name field by clicking the row with the * symbol.
4. In the **Qualifier** field, type **EDI**.
5. In the **Value** field, type **EDI://1234567:ZZ:1234567**.
6. Click **OK**.

You have now created the parties for the new interchange. Party aliases are used to determine the recipient for inbound messages. When a message is received during an exchange, the Base EDI adapter evaluates the logical address of the sender of the message, creates a URI from it, and tries to locate the Receive Location record that matches that URI. The Base EDI adapter then takes the logical address of the recipient of the message, creates a URI from that, and tries to find the matching Party Alias record. When a matched record is found, a message context is created with the sender's URI and recipient's qualifier and value. BizTalk Server then routes the message to the proper recipient.

To create the send ports used to send documents to each party, go to Step 4: Create a Send Port .

Step 4: Create a Send Port

In this section, you continue working within BizTalk Explorer in Visual Studio 2005.

You create a send port that is used to send EDI documents to Northwind. You configure the send port with details relating to the party Northwind, which will be the recipient of the EDI interchange.

To create a send port called XMLToNorthwind

1. Right-click the **Send Ports** folder and then click **Add Send Port**.
2. From the drop-down list, select **Static one-way Port**.
3. In the **Name** field, type **XMLToNorthwind**.
4. In the **Transport Type** field, select **FILE** from the drop-down list.
5. Select the **Address (URI)** field and click the ellipsis (**...**) button.

The File Transport properties page appears.

To configure FILE transport properties

1. In the **Destination folder** field, navigate to **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Northwind\In**.
2. Leave the default options for the filename and copy mode fields.
3. Click **OK**.

To configure the send pipeline value

1. Click the **Send** folder.
2. Expand the **Send Pipeline** property, click the **Send pipeline** field, and select **XMLTransmit** from the drop-down list.
3. Click **OK**.

You have now defined the properties of the send port that will be used to send XML documents from Contoso to Northwind. The send port you just created is a file system-based transport. In this case, XML documents converted from native EDI format will be sent to the folder C:\Documents and Settings\All Users\Application Data\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Northwind\In. To create the receive location that picks up the original EDI document from Contoso, go to Step 5: Create a Receive Port .

Step 5: Create a Receive Port

So far you have created the conduit through which translated XML documents will be sent to Northwind. Now you create a receive location that picks up the original EDI file. You populate the fields of the receive location with values relating to Contoso, which sends the EDI document to Northwind.

You first create a receive port, and then create a receive location for that port.

To create a new receive port

1. Right-click the **Receive Ports** folder, and then click **Add Receive Port**.
2. From the drop-down list, select **One-Way Port**.
3. In the **Name** field, type **EDIfromContoso**.
4. Click **OK**.

To create a new receive location

1. Expand the EDIfromContoso Receive Port tree view, right-click the **Receive Locations** folder, and then click **Add Receive Location**.

2. In the **Name** field, type **PickupEDIfromContoso**.

3. In the **Transport** field, select **EDI** from the drop-down list.

4. Select the **Address (URI)** field and click the ellipsis (**...**) button.

5. The EDI properties screen appears.

To configure the EDI transport properties

1. Expand the Adapter Properties tree view. In the **EDI Address (URI)** folder, click the ellipsis (**...**) button.

2. Select **Contoso**, which has a URI value of EDI://7654321:ZZ:7654321.

3. Expand the **Supported Document Types** for X-12, and then expand the **4010** category.

   The list contains all the available X12 4010 document types.

4. In the **850** field, select the **4010** envelope from the drop-down list.

5. Click **OK**.

To complete configuration of the receive location

1. Click the **Receive Handler** field and select **BizTalkServerApplication** from the drop-down list.

- Click the **Receive Pipelines** field and select **XmlReceive** from the drop-down list.

- Click **OK**.

You have now configured the receive handler to allow X12 850 documents using 4010 envelopes. Note that as part of the receive location configuration you can allow or disallow the types of documents that can be received. This enables you to select only the document type or types that you will accept.

To create an association between the recipient of the EDI message and the send port, go to Step 6: Associate a Send Port with a Party .

Step 6: Associate a Send Port with a Party

So far, you have created the parties involved in the interchange along with the conduits through which the Base EDI adapter delivers EDI documents to Contoso.

Now you link the recipient party Northwind to the XMLtoNorthwind send port.

To associate a party to a send port

1. In BizTalk Explorer, click the **Parties** folder.
2. Right-click **Northwind** and then click **Edit**.
3. Click the **Send Ports** tab and select **XMLtoNorthwind** from the drop-down list.
4. Click **OK**.

To add the document definitions, mapping, and orchestration necessary to translate the document from EDI to XML format, go to the next topic Step 7: Add Document Definitions and Orchestrations

Step 7: Add Document Definitions and Orchestrations

In this step, you add the document formats required by each trading partner (namely XML and X12 4010 850), a map to convert one document format to the other, and an orchestration.

The document definitions, mapping, and orchestration are a part of the Getting started with EDI.sln project, which you opened in Step 2: Open the EDI Solution . The document definitions, mapping, and orchestrations for this interchange are therefore already available. To validate the XML schema, go to the next topic, Step 8: Validate the XML Schema .

Step 8: Validate the XML Schema

In this section, you work within Solution Explorer in Visual Studio 2005.

By validating an XML schema you are converting the XML document used by Northwind into an EDI representation of that XML document. Upon completion of the validation process, the XML schema and an EDI representation of that schema are available from the document definition repository of the Base EDI adapter.

To validate an XSD schema

1. Double-click **X124010850Schema.xsd** to open the schema in the Schema Editor. Do not modify the schema.
2. In the Properties window, select **Schema Editor Extensions** and then click the ellipsis (**...**) button.
3. From the extensions list, select **Covast EDI Schema Editor Extension**.
4. Click **OK**.
5. In Solution Explorer, right-click **X124010850Schema.xsd** and then click **Validate Schema** to validate the schema.

6. When schema validation is complete, you will receive the message "Component invocation succeeded" in the project output.

The schema has now been validated. To continue, go to Step 9: Verify Database Settings .

Step 9: Verify Database Settings

Before you continue with this tutorial, ensure that you are using the correct BizTalk Management database.

To verify database settings

1. In Solution Explorer, right-click **Session 1** and then click **Properties**.

2. In the Session 1 Property Pages, click the **Configuration Properties** folder, and then select **Deployment**.

3. Set the **Server** field to the SQL server where the BizTalk Management database is located.

4. Leave the **Application Name** field blank. This ensures that this project will be deployed to the default BizTalk application.

5. Create an assembly key file by running the following command at a Visual Studio 2005 command prompt:

   sn -k "<*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Visual Studio Projects\Getting Started with EDI\Getting Started with EDI.snk"

6. Click the Common Properties folder and select Assembly.

7. In the Assembly Key File field, click the ellipsis (...) button and navigate to <*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Visual Studio Projects\Getting Started with EDI\Getting Started with EDI.snk.

8. Click **OK**.

   The database settings are now set. To continue, go to Step 10: Restart the BizTalk Services .

Step 10: Restart the BizTalk Services

Before you configure the Base EDI adapter, restart the BizTalkServerApplication service and the BizTalk Base EDI service. This forces the BizTalk Server and Base EDI services to immediately pick up all the changes you made to the solution in the preceding steps.

To restart the BizTalk services

1. Click **Start**, point to **Settings**, and then click **Control Panel**.

2. In Control Panel, double-click **Administrative Tools**, and then double-click **Services**.

3. Right-click **BizTalk Service BizTalk Group : BizTalkServerApplication**, and then click **Restart**.

4. Right-click **BizTalk Base EDI Service**, and then click **Restart**.

   The BizTalk services have now been restarted. To continue, go to Step 11: Configure the Base EDI Adapter .

Step 11: Configure the Base EDI Adapter

You now create the channel through which documents are sent from Contoso to Northwind. This is a one-way channel of communication through which the EDI document, sent from Contoso, is translated by the Base EDI adapter into an XML format and then sent to Northwind.

The Base EDI adapter contains a predefined send handler and receive handler.

To open the Base EDI adapter

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

3. In the expanded adapter list, select **EDI**.

To configure the send handler

1. Double-click the send handler for the EDI adapter.

2. In the **EDI - Adapter Handler Properties** dialog box, click **Properties**.

3. In the **EDI Address (URI)** field, click the ellipsis (**...**) button.

4. Select the party **Northwind** with the URI **EDI://1234567:ZZ:1234567** and click **OK**.

   The URI acts as an identifier for Northwind; it is the code by which Northwind is known in the EDI world. This identification is used within the EDI documents.

5. Expand the **Connection** properties. In the **Port** field, type **11010**.

This is the port number through which documents will be sent to the Base EDI adapter.

6. Click **OK** and **OK** again to complete the configuration of the Base EDI adapter send handler.

To configure the receive handler

1. Double-click the receive handler for the EDI adapter.
2. In the **EDI - Adapter Handler Properties** dialog box, click **Properties**.
3. Expand **Connector Properties**, expand **File System**, and then expand **Account**.
4. In the **Folder (full path or UNC)** field, type **<***drive***>:\Documents and Settings\All Users\Application Data\Microsoft\BizTalk Server 2006\EDI\Subsystem\Documents\PickupEDI**. (Substitute the appropriate value for <*drive*>.)
5. Expand **Connection Properties**, and in the **Port** field, type **11011**.

   This is the port through which BizTalk Server 2006 receives messages from the Base EDI adapter.

6. Click **OK** and **OK** again to complete the configuration of the Base EDI adapter receive handler.

The configuration of the EDI process is now complete. To run the EDI solution, go to Lesson 3: Run the EDI-to-XML Solution .

Lesson 3: Run the EDI-to-XML Solution

You have completed the necessary configuration steps. Now you must prepare to run the solution. To do this, go to Step 1: Deploy the EDI Solution .

In This Section

- Step 1: Deploy the EDI Solution
- Step 2: Bind the Orchestration
- Step 3: Run the Solution

Step 1: Deploy the EDI Solution

This lesson describes the steps that must be completed to deploy the EDI solution to the BizTalk Management database, bind the deployed orchestration to the appropriate ports, and test the solution by processing files.

To deploy the EDI solution

1. In Solution Explorer, right-click **Getting Started with EDI** and then click **Rebuild Solution**.

   The build may take a few moments to complete.

2. After the rebuild is complete, right-click **Getting Started with EDI** again, and then click **Deploy Solution**.

   This deploys the solution to the BizTalk Management database. Again, this will take some time to complete.

After the solution has been deployed, you can bind the deployed orchestration to the appropriate ports and host. Go to Step 2: Bind the Orchestration .

Step 2: Bind the Orchestration

Complete the following steps to bind the deployed BizTalk orchestration to the appropriate BizTalk ports and host instance and start the deployed orchestration so that you can process files.

To bind the orchestration

1. Refresh the BizTalk Explorer view (right-click the Management database and then click **Refresh**), and you will see that the orchestration is now present in the **Orchestrations** folder.

2. In the **Orchestrations** folder, right-click **Getting_Started_with_EDI.OrchestrateContoso850s**, and then click **Bind**.

3. In the **Inbound Ports** field, select **EDIfromContoso** and in the **Outbound Ports - Static** field, select **XMLToNorthwind**.

4. Click the **Host** tab, and then select **BizTalkServerApplication** from the drop-down list.

5. Click **OK**.

6. Right-click **Getting_Started_with_EDI.OrchestrateContoso850s** and then click **Enlist**.

   This process may take a few moments to complete.

7. Right-click **Getting_Started_with_EDI.OrchestrateContoso850s** and then click **Start** to display the **BizTalk Explorer - Express Start** dialog box.

8. Accept all the default options and then click **OK**.

The EDI solution has now been built and started. The next step is to test the solution by processing files. To do this, go to Step 3: Run the Solution .

Step 3: Run the Solution

At this point, all the necessary configuration steps have been completed. All that remains is to run the solution.

However, it is recommended that before running the EDI solution, you first restart the BizTalkServerApplication service and the BizTalk Base EDI service again to ensure that all changes to the solution's configuration are picked up immediately.

To run the EDI solution

1. Navigate to <*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Contoso\Pickup\Sample.

2. Copy the file ContosoPickupInstance.edi.

3. Paste the file into the <*drive*>:\Documents and Settings\All Users\Application Data\Microsoft\BizTalk Server 2006\EDI\Subsystem\Documents\PickupEDI folder.

   The Base EDI adapter picks up the file and translates it into the X-12 4010 850 document format.

   The translated file is then placed in the <*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Northwind\In folder.

This completes the first Base EDI adapter tutorial. To proceed, go to Tutorial 2: XML-to-EDI Document Translation

Tutorial 2: XML-to-EDI Document Translation

This tutorial covers an XML-to-EDI document interchange. The Microsoft BizTalk Server 2006 Base EDI adapter picks up the XML document and then translates it into an EDI document format.

Before you begin these lessons, you should be familiar with the fundamental concepts of the Base EDI adapter and with the tools and processes that you need to start building an EDI solution with BizTalk Server 2006.

Review the following documentation before you begin these lessons:

- Base EDI Adapter Operations
- Developing EDI Business Processes

Requirements

- To successfully complete this tutorial, the Base EDI adapter and all its components must be installed on your computer. For more information, see Installing and Configuring BizTalk Server 2006.

- Ensure that the BizTalk Base EDI service has been started. To do so, click **Start**, point to **Settings**, and then click **Control Panel**. In Control Panel, double-click **Administrative Tools**. In Administrative Tools, double-click **Component Services**.

- The necessary tutorial files can be found at <*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI.

- Ensure that you are part of the EDI subsystem user group before beginning this tutorial. Contact your administrator for details.

- Complete Tutorial 1: EDI-to-XML Document Translation .

User Account Prerequisites

When using the Configuration Wizard, you must configure the BizTalk Base EDI service using the same logon credentials as the BizTalk Server Host service.

To ensure that the user account has sufficient permissions to access the EDI and BizTalk Management databases, the user who configures the EDI send handler must be part of the appropriate NT Users groups, namely the EDI Subsystem and BizTalk Server Administration user groups, respectively.

To access the BizTalk Explorer Object Model as a non-administrator, do the following:

1. Add the EDI Subsystem Users group to the BTS_HOST_USERS role found in the BizTalkMgmtDb BizTalk Management database.

2. When developing EDI solutions ensure that:

    - The logged-on user is a part of the EDI users group.

    - If the EDI service is running as a local machine account, you must log on and develop using a local machine account.

In This Section

- Lesson 1: Plan the XML-to-EDI Solution

- Lesson 2: Configure the XML-to-EDI Solution

- Lesson 3: Run the XML-to-EDI Solution

- Lesson 4: Acknowledge the Interchange

Lesson 1: Plan the XML-to-EDI Solution

BizTalk Server 2006 provides a development and execution environment for exchanging documents using electronic data interchange (EDI) both within and between businesses.

EDI Scenario

This tutorial is a continuation of the EDI-to-XML solution, Tutorial 1: EDI-to-XML Document Translation .

In this tutorial, you develop an XML-to-EDI solution. In this solution, Northwind sends an XML document to Contoso and receives confirmation of receipt of the document from Contoso in the form of a 997 functional acknowledgment document.

To begin configuring the solution, go to Lesson 2: Configure the XML-to-EDI Solution .

Lesson 2: Configure the XML-to-EDI Solution

In this lesson you configure an EDI solution that enables two trading partners to exchange data, even though they use different formats for the transfer of electronic data.

In This Section

- Step 1: Start the Base EDI Service
- Step 2: Create a Send Port
- Step 3: Create a Receive Port
- Step 4: Associate a Send Port with a Party
- Step 5: Enable the Receive Location
- Step 6: Enlist and Start the Send Port
- Step 7: Cycle the Services

Step 1: Start the Base EDI Service

By default, the BizTalk Base EDI service is set to Disabled. You must start this service to run the EDI solution.

To start the BizTalk Base EDI service

1. Click **Start**, point to **Settings**, point to **Administrative Tools**, and then click **Services**.
2. Right-click **BizTalk Base EDI service**, and then click **Properties**.

3. In the **Startup type** dialog box, select **Manual** from the drop-down list.

4. Right-click **BizTalk Base EDI service**, and then click **Start**.

To begin configuring the XML-to-EDI solution, go to Step 2: Create a Send Port .

Step 2: Create a Send Port

In this section, you create a send port that is used to send documents converted from XML to native EDI format to Contoso. You configure the send port with details about the party Contoso, the recipient of the document.

To create a send port called EDItoContoso

1. Right-click the **Send Ports** folder and then click **Add Send Port**.

2. From the drop-down list, select **Static one-way Port**.

3. In the **Name** field, type **EDItoContoso**.

4. In the **Transport Type** field, select **EDI** from the drop-down list.

5. Select the **Address (URI)** field and click the ellipsis (**...**) button.

   The EDI Properties page appears.

To configure EDI transport properties

1. Expand **Adapter Properties**. In the **Logical address** field, click the ellipsis (**...**) button and select **Contoso**. The value in the URI field will be **EDI://7654321:ZZ:7654321**.

2. Expand **Connector properties**. In the **Connector** field, select **File System** from the drop-down list.

3. Expand the **File system** property and type **edi#.txt** in the **File mask** field.

   This specifies that all XML files converted to EDI format will be given the extension ".edi". The "#" character will be replaced during run time with the interchange number in the file.

4. In the **Folder (full path or UNC)** field, type the path **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Contoso\Out**.

   This specifies the file system location where the documents converted from XML to EDI will be sent.

5. Expand the **Supported Document Types for X12** property, and then expand the **X12-4010** category. In the **850** field, select the **4010** envelope.

6. Click **OK** to continue and return to the static one-way send port properties.

You must now specify a send pipeline for this port.

To configure the send pipeline value

1. Click the **Send** folder.
2. Expand the **Send Pipeline** property, click the **send pipeline** field, and select **XMLTransmit** from the drop-down list.
3. Click **OK**.

You have created the send port through which a translated EDI document (in this case an X-12 4010 850) will be sent to Contoso. Note that as part of the send port configuration you can allow or disallow the types of documents that can be sent. This enables you to specify the type or types of documents that you will allow.

To create the receive location that will pick up the original XML document from Northwind, go to Step 3: Create a Receive Port .

Step 3: Create a Receive Port

So far you have created the conduit through which translated EDI documents are sent to Contoso. Now you create a receive location to pick up the original file. You populate the fields of the receive location with values relating to Northwind, which sends the XML document to Contoso.

First you must create a receive port.

To create a new receive port

1. Right-click the **Receive Ports** folder, and then click **Add Receive Port**.
2. From the drop-down list, select **One-way Port**.
3. In the **Name** field, type **XMLfromNorthwind**.
4. Click **OK**.

The receive port is created. You now create a receive location for this receive port.

To create a new receive location

1. Expand the **XMLfromNorthwind** receive port, right-click the **Receive Locations** folder, and then click **Add Receive Location**.
2. In the **Name** field, type **PickupXMLfromNorthwind**.
3. In the **Transport** field, select **File** from the drop-down list.

4. Select the **Address (URI)** field and click the ellipsis (**...**) button.

This brings up the File transport properties page.

To configure the file transport properties

1. In the receive folder field, enter the source location for the XML document that is sent from Northwind to Contoso: **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Northwind\Pickup**.
2. In the **File mask** field, type ***.xml**.
3. Click **OK**.

To complete configuration of the receive location

1. Click the **Receive Handler** folder and select **BizTalkServerApplication** from the drop-down list.
2. Click the **Receive Pipelines** folder and select **XMLReceive** from the drop-down list.
3. Click **OK**.

The receive location that will be used to receive XML documents from Northwind is now configured. Go to Step 4: Associate a Send Port with a Party .

Step 4: Associate a Send Port with a Party

So far you have created the parties involved in the interchange along with the conduits through which the Base EDI adapter delivers EDI documents to Contoso.

Now you link the recipient party, Contoso, to the EDItoContoso send port.

To associate a party to a send port

1. In BizTalk Explorer, click the **Parties** folder.
2. Right-click **Contoso** and then click **Edit**.
3. Click the **Send Ports** tab and select **EDItoContoso** from the drop-down list.
4. Click **OK**.

Step 5: Enable the Receive Location

You now enable the PickupXMLfromNorthwind receive location.

To enable the receive location

1. Right-click **PickupXMLfromNorthwind**, and then click **Enable**.

One step remains in the configuration of the EDI process.

Step 6: Enlist and Start the Send Port

You now enlist the EDItoContoso send port.

To enlist a send port

1. Right-click **EDItoContoso**, and then click **Enlist**.
2. Right-click **EDItoContoso**, and then click **Start**.

You have completed the configuration of the send port and receive locations. Before running the EDI solution, restart the BizTalkServiceApplication and BizTalk Base EDI services.

Step 7: Cycle the Services

Before you configure the Base EDI adapter, you first restart the BizTalk services.

To restart the BizTalk services

1. Click **Start**, point to **Settings**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**, and then double-click **Services**.
3. Right-click **BizTalk Service BizTalk Group : BizTalkServerApplication**, and then click **Restart**.
4. Right-click **BizTalk Base EDI Service**, and then click **Restart**.

Lesson 3: Run the XML-to-EDI Solution

You have completed the necessary configuration steps. Now you must prepare to run the solution.

In This Section

- Step 1: Bind the Orchestration
- Step 2: Run the Solution

Step 1: Bind the Orchestration

Complete the following steps to bind the deployed BizTalk orchestration to the appropriate BizTalk ports and host instance and start the deployed orchestration.

To bind the orchestration

1. Refresh the BizTalk Explorer view, right-click the Management database, and then click **Refresh**.

   The orchestration is now present in the Orchestrations folder.

2. Right-click the **OrchestrateNorthwindreqs** orchestration, and then click **Bind**.

3. In the **Inbound Ports** field, select **XMLfromNorthwind**, and in the **Outbound Ports - Static** field, select **EDItoContoso**.

4. Click the **Host** tab, and then select **BizTalkServerApplication** from the drop-down list.

5. Click **OK**.

6. Right-click **OrchestrateNorthwindreqs** and then click **Enlist**.

   It may take a few moments to complete this process.

7. Right-click **OrchestrateNorthwindreqs** and then click **Start**. Accept all the default options.

8. Click **OK**.

Step 2: Run the Solution

At this point, all the necessary configuration steps have been completed. All that remains is to run the solution.

It is recommended that you first restart the BizTalk services and the BizTalk Base EDI component services.

To run the EDI solution

1. Browse to **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Northwind\Pickup\Sample**.

2. Copy the file **NorthwindPickupInstance.xml**.

3. Paste the file into the **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Northwind\Pickup** directory.

The following actions occur:

a. The **PickupXMLfromNorthwind** receive location picks up the file.

b. The **PickupXMLfromNorthwind** receive location is bound to the **XMLfromNorthwind** receive port, and the **XMLfromNorthwind** receive port is bound to the **Getting_Started_with_EDI.OrchestrateNorthwindReqs** orchestration. This orchestration maps the document into the X-12 4010 850 document format and sends it to the **EDItoContoso** send port.

c. The **EDItoContoso** send port specifies a transport type of EDI and an address of EDI://7654321:ZZ:7654321, so the document is passed to the EDI subsystem for processing.

d. The translated file is then placed in the **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Contoso\Out** directory because the **EDItoContoso** send port specifies this directory in its Connector properties.

To send a functional acknowledgment for this interchange

Lesson 4: Acknowledge the Interchange

You have completed the configuration for EDI services and run the EDI solution.

A common practice within EDI interchanges is for an acknowledgment message to be sent upon receipt of the interchange. This message states that the interchange was processed successfully, or if not, details the errors that occurred, enabling the original sender of the interchange to reformat the interchange and resend it.

In this scenario Contoso acknowledges that it received the X-12 850 document from Northwind by sending an X-12 997 functional acknowledgment to Northwind.

In This Section

- Step 1: Open Health and Activity Tracking
- Step 2: Edit the EDI File
- Step 3: Modify the Send Port
- Step 4: Run the Solution

Step 1: Open Health and Activity Tracking

The Health and Activity Tracking tool provides information about all processes applied to the EDI document that Contoso received. After you start Health and Activity Tracking, you can view the EDI details of the X-12 document sent to Contoso.

To start Health and Activity Tracking

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Health and Activity Tracking**.

To view EDI details

1. On the **Reporting** menu, click **EDI Reports**.
2. In the **Direction** field, select **Send**.
3. In the **Status** field, select **Sent**.
4. In the **Schema** field, select **850/004/010/DEFAULT/X**.
5. Click **Run Query**.
6. Right-click the first document returned by the query, and then click **EDI Details**.
7. Click the **References** tab and copy the values in the **Group reference** and **Document Reference** fields.

You copied the values in the **Group reference** and **Document reference** fields to paste these values into the 997 functional acknowledgment.

Step 2: Edit the EDI File

You are going to copy the values found in the **Group reference** and **Message reference** fields of the X-12 document sent to Contoso into the 997 functional acknowledgement document.

By doing this, you ensure that when the Base EDI adapter picks up the 997 document, it recognizes that this document corresponds to the original X-12 850 document sent from the Northwind interchange to Contoso.

To edit the EDI file

1. Browse to the **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Contoso\Pickup\Sample** directory.
2. Open the **ContosoPickup997.edi** file with Notepad.
3. Replace the string **GCR** with the value that you copied from the **Group reference** field in the Health and Activity Tracking tool.

4. Replace the string **MRN** with the value that you copied from the **Document reference** field in the Health and Activity Tracking tool.

5. Save and close the modified **ContosoPickup997.edi** file.

Step 3: Modify the Send Port

You now modify the Configuration properties for the EDItoContoso send port. This indicates to Contoso that Northwind always expects a functional acknowledgment.

The reason you modify the properties of the send port now, and not during the initial configuration stages, is to ensure that time-out errors do not occur in the first stage.

To configure the send port for functional acknowledgements

1. In BizTalk Explorer, right-click **EDItoContoso** and then click **Edit**.

2. In the **Address (URI)** field, click the ellipsis (**...**) button.

3. Expand the Flags tree view in the EDI transport properties.

4. Set the **Functional Acknowledgement** field to **Always**.

5. Click **OK**.

The configuration of a functional acknowledgment has now been completed.

Step 4: Run the Solution

All the necessary configuration steps have been completed. All that remains is to run the solution.

It is recommended that you first restart the BizTalk services and BizTalk Base EDI component services.

To send the functional acknowledgement

1. Browse to **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\Getting Started with EDI\Contoso\Pickup\Sample**.

2. Copy the edited file **ContosoPickup997.edi**.

3. Paste the file into **<***drive***>:\Documents and Settings\All Users\Application Data\Microsoft\BizTalk Server 2006\EDI\Subsystem\Documents\PickupEDI**.

The Base EDI adapter now picks up the file. You can check the status of the original outbound X-12 850 document again by using Health and Activity Tracking.

The **Received** field under **Functional Acknowledgement** on the **Acknowledgements** tab will be updated to **Yes**.

This completes the XML-to-EDI Base EDI tutorial.

Base EDI Adapter Operations

This section contains information to enable you to administer, operate, and maintain the Microsoft BizTalk Server 2006 Base EDI adapter.

First ensure that you are part of the EDI subsystem user group. Contact your system administrator for details.

Base EDI Adapter Administration

This section describes how to administer your core Base EDI adapter implementation.

You use the BizTalk Server Base EDI Administration console to:

- Set the numbers of interchanges and documents entering and leaving the Base EDI adapter.
- Configure the characteristics of the trace functionality.
- Configure the characteristics of the archive and delete functionality.
- Customize the performance settings of the Base EDI adapter, such as the number of translation threads that can occur simultaneously.

Using the Base EDI Administration console, you maintain the global system parameters of the Base EDI adapter. These parameters define the characteristics of the functionality of the adapter—for example, how often BizTalk Server should check if there are documents that need to be sent, or how long BizTalk Server should wait to receive a functional acknowledgment document.

These parameters are defined during the installation process, but they can be modified manually.

Base EDI Adapter Operations Management

This section describes how to monitor your Base EDI adapter environment. This day-to-day monitoring is done through Health and Activity Tracking (HAT). HAT enables you to track each document entering the EDI subsystem from the back-end system or an external connector, along with each process that has been applied to the document.

In This Section

- Base EDI Adapter Administration
- Base EDI Adapter Operations Management

Base EDI Adapter Administration

The Microsoft BizTalk Server 2006 Base EDI adapter comes with the Base EDI Administration console. You use the Base EDI Administration console to monitor your global EDI system parameters.

This section provides procedural and conceptual information that enables you to administer your Base EDI adapter environment.

Ensure that you are part of the EDI subsystem user group before using the Base EDI Administration console. Contact your system administrator for details.

The main tasks that the Administration console performs include:

- Tracking the sequence numbers of documents and interchanges.
- Enabling you to customize the specifics of the trace file.
- Enabling you to configure the performance settings of the Base EDI adapter.

In This Section

- How to Use the Base EDI Administration Console
- Numbers Parameters
- Time-Out Parameters
- Tuning Parameters
- Trace Parameters
- Audit Trail Parameters
- Client/Server Parameters

How to Use the Base EDI Administration Console

The BizTalk Server Base EDI Administration console enables you to administer, operate, and maintain the Microsoft BizTalk Server 2006 Base EDI adapter.

The parameters of the Base EDI adapter are set during installation, but can be modified by using the Base EDI Administration console.

To launch the Base EDI Administration console

1. Browse to the **<***BizTalk_installation_directory***>\EDI\Subsystem** directory.
2. Double-click **EDIBTSmmc.msc**.

After you start the Base EDI Administration console, complete the following steps.

To configure the parameters of the Base EDI Administration console

1. Expand the **Microsoft BizTalk Server Base EDI Adapter** folder.
2. Click the **Parameters** folder.
3. Right-click the server listed, and then click **Properties**.

Numbers Parameters

The numbers parameters are internal numbers allocated by the BizTalk Server 2006 Base EDI adapter to each document that enters and leaves the Base EDI adapter.

Use this	To do this
Next Interchange Number	This is the number that will be allocated to the next incoming interchange. The Base EDI adapter automatically allocates and increments this value. It is recommended that you do not manually alter this value.
Next Group Number	This is the number that will be allocated to the next incoming group. The Base EDI adapter automatically allocates and increments this value, It is recommended that you do not manually alter this value.
Next Document Number	This is the number that will be allocated to the next incoming document. The Base EDI adapter automatically allocates and increments this value. It is recommended that you do not manually alter this value.

Time-Out Parameters

The time-out parameters determine how much time can elapse before the Base EDI adapter expects a certain action to occur. For example, you can specify the length of time the Base EDI adapter waits for a functional acknowledgement.

Use this	To do this
Functional Acknowledgements	Enter the maximum number of minutes that can elapse before a functional acknowledgement should be received regarding a document you have sent.

Tuning Parameters

The tuning parameters enable you to fine-tune the operations of the BizTalk Server 2006 Base EDI adapter. Many of these parameters refer to memory usage—for example, the cache to allocate to each partner, or the maximum number of translation threads that can run concurrently.

How best to tune the parameters described below depends on the structure of your deployment and the types of operations it performs. In most cases you will have to experiment with the settings to tune your deployment for optimal performance.

The tuning parameters are managed from three different locations: from the EDIBTSmmc.msc snap-in console (accessible from the <*drive*>\Program Files\Microsoft BizTalk Server 2006\EDI\Subsystem folder), from the BizTalk Server 2006 Administration Console, and from the Windows registry.

Use this	To do this
Check send queue every	Enter the duration of a cycle (in seconds). The Base EDI adapter checks the send queue for outbound documents every X seconds. The Base EDI adapter usually waits some time before it starts sending documents (to reduce communication cost).
Maximum Mapping errors	Enter the value for the maximum number of translation errors the Base EDI adapter can report for one interchange. Setting this number higher than 1 means that the Base EDI adapter does not stop checking a document after the first error has been detected.
Cache Partners	Enter the number of partners you want to cache. The Base EDI adapter uses the cache to speed up access to partner details. You should set this to the number of partners you communicate with, although it consumes some memory. The Base EDI adapter uses a statistical caching mechanism. The most frequently used partners are most likely to stay in the cache.
Cache Custom Numbers	Enter the number of custom numbers you want to cache. The Base EDI adapter uses the cache to speed up access to custom number details. The Base EDI adapter uses a statistical caching mechanism. The most frequently used custom numbers are most likely to stay in the cache.
Maximum Number of translate threads	(EDIBTSmmc snap-in) Specifies the number of concurrent translation threads that are available to translate to and from native EDI format. The more concurrent translation threads you make available, the faster translations will be performed when handling large numbers of documents. You should configure this setting in accordance with the amount of available memory, your processor speed, and the maximum number of files that can be open at the same time (each

	translate session opens two files).
Database pool size	(EDIBTSmmc snap-in) Specifies the number of simultaneous connections that can be used by the BizTalk EDI server to connect to the EDI database.
BatchSize	(BizTalk Server Administration Console) Specifies the maximum number of documents that will be sent from the Base EDI adapter to the EDI subsystem in a single session. Setting this to a higher number reduces processing overhead in BizTalk Server, thus increasing the speed at which documents are sent. However, setting BatchSize to a large number reduces the effect of concurrent transmissions, which can slow down communications.
Sessions	(BizTalk Server Administration Console) Specifies the number of concurrent transmitting sessions that will be available for sending documents from the Base EDI adapter to the EDI subsystem. The more concurrent transmitting sessions that are available, the faster documents will be passed between the Base EDI adapter and the EDI subsystem.
BTSSubmit	(Windows Registry - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EDI Subsystem for Microsoft BizTalk Server\6.0) Specifies the number of concurrent transmitting sessions that will be available to send documents from the EDI subsystem to the Base EDI adapter. The more concurrent transmitting sessions that are available, the faster documents will be passed between the EDI subsystem and the Base EDI adapter.
FileFetch	(Windows registry) Specifies the number of concurrent file connector sessions that will pick up documents from the file system. The more concurrent file connector sessions that are available, the faster documents will be picked up and processed from the file system.
FileSubmit	(Windows registry) Specifies the number of concurrent file connector sessions that will submit documents to the file system. The more concurrent file connector sessions that are available, the faster documents will be processed and submitted to the file system.
FileSubmitBatchSize	(Windows registry) Specifies the number of files that will be sent in a single session by the file connector. Setting this to a higher number may reduce overhead in connecting to file systems, thus speeding up transmissions. However, setting FileSubmitBatchSize to a large number reduces the effect of concurrent transmissions, which can slow down communications.

Trace Parameters

This section provides information about configuring the trace parameters. The generated trace file can be used for analyzing communication sessions or load and download sessions.

Use this	To do this
Trace file	Type the name of the trace file.
Maximum size	Type the maximum size of an individual trace file.
Number of trace files	Type the number of trace files to be created.
Trace kernel processes	Indicate if you want to trace the Base EDI adapter. Normally the trace is turned off. You should turn on the trace function only when you have problems mapping documents, communicating, and so on, and you do not know why there are problems. **Note** This trace is very extensive and the resultant trace file grows rapidly.

Audit Trail Parameters

This section provides information about configuring the audit trail parameters. Before the Base EDI adapter deletes interchanges and documents, it archives them.

Use this	To do this
Automatic	Indicate if you want to automate the archive and delete function. The archive and delete function will be started in the background.
Start every	Select when the automatic archive and delete function should be started. Select from Day, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday.
Start time	Type the time when the automatic archive and delete function should start (on the day entered above).
Process up to	Type the number of days ago the automatic archive and delete function should delete up to.
Archive before delete	Indicate if the interchanges should be archived before they are deleted from the database and hard disk.

The archive function creates archive files, which contain all the information relating to a single interchange—for example, the audit trail information, the original interchange, error information, and the translated documents.

Client/Server Parameters

The client/server parameters provide information about the settings of the server on which the Base EDI adapter is running.

Use this	To do this
Server hostname	Type the name of the server on which the Base EDI adapter is running (as known within the Transmission Control Protocol/Internet Protocol (TCP/IP) network).
Server Portname/Number	Type the port number for this server.

Base EDI Adapter Operations Management

Microsoft BizTalk Server 2006 Base EDI adapter operations management enables you to monitor the status of each document that the Base EDI adapter processes.

This section provides procedural and conceptual information to assist you in monitoring the processing of documents through the Base EDI adapter.

Ensure that you are part of the EDI subsystem user group before using Base EDI adapter operations management. Contact your system administrator for details.

The main tasks performed by Base EDI adapter operations management are:

- Track the status of all documents sent or received by the Base EDI adapter.
- Track the status of functional acknowledgements.
- Provide sender, recipient, and interchange information at a glance.

In This Section

- How to Use HAT with the Base EDI Adapter
- General Section
- References Section
- Acknowledgements Section
- Details Section
- Updating the EDI Codelist Database and EDI Engine Input File When Visual Studio 2005 Is Not Installed

How to Use HAT with the Base EDI Adapter

Health and Activity Tracking (HAT) for EDI provides you with information to monitor the flow of documents through the Base EDI adapter. You get Health and Activity Tracking information about individual documents along with the group and interchange to which each document belongs.

To launch Health and Activity Tracking for EDI

1. Click **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Health and Activity Tracking.**

To view the EDI details of a document

1. On the **Reporting** menu, click **EDI Reports**.
2. In the **Direction** field, select **Receive** or **Send**.
3. Click **Run Query**.
4. Right-click the required document, and then click **EDI Details**.

This provides you with the EDI specifications for the selected document.

General Section

The General section provides overview information about the document, such as the generic type, format, and the specifics of how the Base EDI adapter is processing the document.

Use this	To do this
Number	This field shows the number of the document, which the Base EDI adapter allocates.
Group	This field shows the number of the group to which the document belongs, which the Base EDI adapter allocates to the document.
Interchange	This field shows the number of the interchange to which the document belongs, which the Base EDI adapter allocates.
Format	This is the format description of the document, showing the status of the document in an EXTERNAL format or an INTERNAL format.
Type (version and release)	This is the current type, version, and release of the received document, showing the status of the document in an EXTERNAL type or an INTERNAL type.
Direction	This field indicates whether this document was sent or received.

Status	This is the current status of the document, which can be one of the following: • In external format • In internal format • Plug-in is processing • Downloaded

References Section

The References section provides sender and recipient information about the document along with reference numbers for the document group and interchange.

Use this	To do this
MessageID	BizTalk Server generates this message after translations. It is used for internal administration purposes.
Interchange reference	This is the interchange control reference as stored in the document. For example, for EDIFACT and ANSI X-12 documents, it is retrieved from the UNB and ISA segments respectively.
Group Reference	This is the group reference number stored in the document.
Document reference	This is the document reference number as stored in the document. For example, for EDIFACT and ANSI X-12 documents, it is retrieved from the UNH and ST segments respectively.
Sender URI	This field identifies the sender of the document. Within the EDIFACT and ANSI X-12 EDI standards, the sender is identified by an element in the UNB and ISA segment respectively. Behind the sender identification, the sender identification code qualifier is displayed (if used).
Recipient URI	This field identifies the recipient of the document. In EDIFACT and ANSI X-12 documents, this is an element of the UNB and ISA segment respectively. Behind the recipient identification, the recipient identification code qualifier is displayed (if used).

Acknowledgements Section

This section provides information about the status of functional acknowledgments relating to the document, along with the group and interchange to which the document belonged.

Use this	To do this
Sent	This field indicates whether the Base EDI adapter has acknowledged this document.
Interchange number	This field indicates the interchange number of the functional acknowledgment document.
Group Number	This field indicates the group number of the functional acknowledgment document.
Document number	This field indicates the document number of the functional acknowledgment document.

Details Section

The Details section provides specifics about the translation process, indicating the transport through which the document was received, the size of the file, and the times that the file was both sent and received.

Use this	To do this
External Connector	This field indicates the external connector through which this document was received.
Number of bytes	The field indicates the size of the document in bytes (characters).
Test indicator	This is the test indicator as stored in the EDIFACT and ANSI X-12 document (in the UNB and ISA segment respectively). This is used to identify an EDIFACT and ANSI X-12 interchange as a test interchange. The business database application that retrieves this document should not store this document in the operational database, but only use it as a test. This indicator will therefore also be stored in the envelope.
Entered on	This field indicates the date and time that the Base EDI adapter received this document. For all documents in the interchange, this is the date and time of the interchange.
Translated on	The field indicates the date and time that the Base EDI adapter processed this document.

Left on	This field indicates the date and time that the document left the Base EDI adapter.
Show EDI Message	Clicking this button opens Notepad and displays the content of the EDI message.

Updating the EDI Codelist Database and EDI Engine Input File When Visual Studio 2005 Is Not Installed

You can update the electronic data interchange (EDI) codelist database and EDI Engine Input file with one or more EDI schemas by using command-line tools. The EDI codelist database stores the schemas used by the EDI runtime. The EDI Engine Input file is used by the EDI subsystem at run time and contains a copy of the schemas in the codelist database.

This functionality is provided to update the EDI system on a BizTalk Server computer that does not have Visual Studio 2005 installed.

Using the XSD2EDI.exe and COMPEIF.exe Command-Line Tools

XSD2EDI

To manually update the EDI codelist database (EDICodeLists.mdb) with an EDI schema, run the XSD2EDI command at a command prompt:

XSD2EDI -c[Location of EDICodelists.mdb] [Base EDI schema

For example, run the XSD2EDI command at a command prompt if the EDICodeLists.mdb file is in the default directory and you are updating the BizTalk EDI database with the X12 4010 850Schema.xsd schema file:

XSD2EDI -c"C:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\CodeLists\EDICodeLists.mdb" "C:\Program Files\Microsoft BizTalk Server 2006\EDI\Adapter\EDI Schemas\X12\4010\850Schema.xsd"

XSD2EDI validates the specified schema before updating the specified EDI codelist database. XSD2EDI.exe uses the following parameters:

Usage: **xsd2edi inputfile [-c | -f | -o | -r | -s | -v]**

Where:

inputfile = file path of XSD to convert or remove

-cfilepath = file path to Access database containing EDI codes

default: **%ESP_SRV%\..\Adapter\CodeLists\EDICodeLists.mdb**

-f = convert in fastmode; that is, do not compile the repository

-r = remove inputfile from the repository (do not convert)

-s = save existing version of definition

default: turned off

-vtmpfile = temp file to store output for Visual Studio 2005

COMPEIF

To manually update the EDI Engine Input file, run the COMPEIF.exe command at a command prompt:

COMPEIF

The COMPEIF tool merges the contents of the EDI codelist database into the EDI Engine Input file, and therefore it should be executed after updating the codelist database with XSD2EDI.

After the COMPEIF command has completed, restart the BizTalk Base EDI service on the destination computer.

Configuring the Base EDI Adapter

This section provides instructions for configuring the Base EDI adapter.

When EDI is selected as a transport type, documents are sent through the Base EDI adapter. You can configure the EDI transport to the needs of specific business partners; for example, you can specify that only certain types of document formats can be sent through the EDI transport.

Ensure that you are part of the EDI subsystem user group before configuring the EDI transport. Contact your system administrator for details.

In This Section

- Base EDI Adapter Configuration Prerequisites
- How to Configure a Base EDI Receive Handler
- How to Configure a Base EDI Receive Location
- How to Configure a Base EDI Send Handler
- How to Configure a Base EDI Send Port

Base EDI Adapter Configuration Prerequisites

This section contains information about configuration prerequisites for the BizTalk Server Base EDI adapter.

User Account Prerequisites

When using the Configuration Wizard, you must configure the BizTalk Base EDI service using the same logon credentials as the BizTalk Server Host service.

To ensure that the user account has sufficient permissions to access the EDI and BizTalk Management databases, the user who configures the EDI send handler must be part of the appropriate NT Users groups, namely the EDI Subsystem and BizTalk Server Administration user groups.

To access the BizTalk Explorer Object Model as a non-administrator, do the following:

1. Add the EDI Subsystem Users group to the BTS_HOST_USERS role found in the BizTalkMgmtDb BizTalk Management database.

2. When developing EDI solutions ensure that:

   - The logged-on user is a part of the EDI users group.

   - If the EDI service is running as a local machine account, you must log on and develop using a local machine account.

Security Prerequisites

The BizTalk Base EDI service runs using a non-administrator NT account. Ensure that adequate permissions have been provided for the file pick-up and drop locations (Read/Write) at run time. Contact your system administrator for details.

When you do not have sufficient permissions, you will get "Permission denied" errors in the Event Viewer.

Registry Prerequisites

When the BizTalk Server Administration and run-time components are installed on separate computers, you need to add the following registry entries on the Administration computer only.

The registry keys to add are the following:

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EDI Subsystem for Microsoft BizTalk Server\4.0] "Database"="MyBizTalkEDIDB" (where MyBizTalkEDIDB is the name of the EDI database)

and

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EDI Subsystem for Microsoft BizTalk Server\4.0] "Database"="DatabaseServer"="RemoteSQLHost" (where RemoteSQLHost is the name of the SQL server)

This registry entry enables you to access an EDI database that does not use the default name of **BiztalkEDIDb** on a remote SQL server.

When this registry entry is absent, however, BizTalk Server tries to connect to the default EDI database on the same SQL server where the BizTalk Management database is present.

How to Configure a Base EDI Receive Handler

The receive handler sends documents to BizTalk Server. When configuring the receive handler, you specify the port from which the BizTalk Server 2006 Base EDI adapter sends messages to BizTalk Server.

Use the **Receive Handler Properties** dialog box to configure the receive handler properties for the Base EDI adapter.

To configure a Base EDI receive handler

1. In the **Adapter Handler Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Host name	Select the host with which the receive handler will be associated. BizTalkServerApplication is the default host for the Base EDI adapter.

2. On the **Properties** tab, do the following:

Use this	To do this
Trace EDI Adapter	This is a mandatory field. Specify the level of information to be written to the Event Log. Options: • Error • Info • Off

	• Verbose • Warning
Port	This is a mandatory field. Specify the port the EDI receive handler uses internally.
File Mask	This is a mandatory field. Specify the mask for the files. This mask can contain the standard wildcard value "*.". The sign is replaced at run time with the interchange number. Default value: *.edi Type: String
Folder	This is a mandatory field. Specify the directory location for the receipt of inbound EDI files.
Password	This is an optional field. Specify the password for the account. You will use this when you want to access a network location.
User Name	This is an optional field. Specify the user name for the account. You will use this when you want to access a network location.
Minutes off	This is an optional field. Specify the number of minutes the File connector is inactive between communication sessions.

How to Configure a Base EDI Receive Location

You can set Base EDI receive location adapter variables in the BizTalk Server Administration console.

To configure variables for a Base EDI receive location

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application you want to create a receive location in.

2. In the BizTalk Server Administration console, click the **Receive Port** node in the left pane. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

3. In the **Receive Port Properties** dialog box, in the left pane, select the **Receive Locations** option, and in the right pane, double-click an existing receive location or click **New** to create a new receive location.

4. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **EDI** from the drop-down list, and then click **Configure** to configure the transport properties for the receive location.

5. In the **EDI Transport Properties** dialog box, do the following:

Adapter properties

In this section, select the logical address that you associate with the receive location.

Use this	To do this
EDI Address (URI)	The EDI subsystem uses logical addresses to determine the sender and recipient of a document. To determine the sender of a document, the EDI subsystem takes the logical address of the sender of a message, creates a URI from that, and tries to find the receive location record with that URI. **Note** The URI for a send port or receive location cannot exceed 256 characters..

Flags

In this section, specify the functional acknowledgment settings for this trading partner.

Use this	To do this
Functional Acknowledgements	The EDI transport requires you to select a setting for functional acknowledgements. Options: • Always • Never

Supported document types

In this section, specify whether to accept unlisted document types.

Use this	To do this
Accept all unlisted documents	Specify whether the EDI adapter will accept documents that are not in the list of supported document types at this receive location. Options: • No • Yes

Supported document types for EDIFACT

Select from the list of supported EDIFACT document types.

Supported document types for X-12

Select from the list of supported X-12 document types.

6. Click **OK**.

7. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and then click **OK** to save the settings. For information about the **Receive Locations Properties** dialog box, see How to Create a Receive Location.

How to Configure a Base EDI Send Handler

The send handler receives documents from BizTalk Server. When configuring the send handler, you specify the port number BizTalk Server uses to send documents to the BizTalk Server 2006 Base EDI adapter.

To specify the port number BizTalk Server uses to send documents to the Base EDI adapter

1. In the **Adapter Handler Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Host name	Select the host with which the send handler will be associated.

2. On the **Properties** tab, do the following:

Use this	To do this
EDI Address (URI)	Required. Select the logical address to associate with the send handler.
Trace EDI Adapter	Required. Specify the level of information to be written to the Event Log. Options: • Error • Info • Off

	• Verbose • Warning
Port	Specify the port the EDI send handler will use internally.

How to Configure a Base EDI Send Port

When configuring the send port, you can enter the specific EDI details relating to your trading partner, for example, the logical address of the recipient or the document format the EDI message must be sent in.

To configure a Base EDI send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. Configure all of the send port options. On the **General** tab, in the **Transport** section, for the **Type** option, specify **EDI**.
2. On the **General** tab, in the **Transport** section, next to **Type**, click **Configure**.
3. In the **EDI Transport Properties** dialog box, do the following:

Adapter properties

Use this	To do this
EDI Address	The EDI subsystem uses EDI addresses to determine the sender and recipient of a document. To determine the recipient, the EDI subsystem takes the EDI address of the recipient of a message, creates a URI from that, and using this URI tries to find the Party Alias record. The EDI address must conform to the EDI syntax; EDI in the **Name** field, EDI in the **Qualifier** field and **Value** must be of the following structure: EDI://aaaa:bb:zzzzz

Connector properties

In the **Connector properties** section, you can configure the specifics of the connector used by the Base EDI adapter, for example the pick-up directory and the file mask used.

Use this	To do this
Connector	The File connector is the default selection when you are using the EDI transport.
File Mask	This is a required field. A mask is a filter that selectively includes or excludes certain values. In this field, * / and ? can be used as wildcards.
Folder (full path	This is the full path of the directory location from which the File connector

or UNC)	picks up files.
Password	Type the password of the shared directory or mapped drive specified in the full path.
Username	Type the user name for the shared directory or mapped drive specified in the full path.

Flag properties

In this section you configure the functional acknowledgment settings.

Use this	To do this
Functional Acknowledgements	The EDI transport requires you to select a setting for functional acknowledgments. Options: • Always • Never

Options properties

In this section you can specify syntax requirements for the document format that will be sent.

Use this	To do this
Component Separator	This field is automatically set to default. Select the separator used for separating components of composite elements. For the EDIFACT UNOA syntax, for example, this would be the colon, and in ANSI X-12 this would be the greater-than (>) sign.
Decimal Notation	This field is automatically set to default. Here you have to indicate the type of decimal notation to be used for this type of document. Enter or select comma, point, or implicit. The setting defined here overrules the default setting for the business document format versions. The decimal notation defined per trading partner in its turn overrules this setting.
Element Separator	This field is automatically set to default. Here you can select the separator used for separating (singular) elements. For the EDIFACT UNOA syntax, for example, this would be the plus (+) sign, and in ANSI X-12 the asterisk (*).

Release character	This field is automatically set to default. Here you can select the character used for releasing (escaping) special characters. This is used to be able to send special characters that are normally separators. For the EDIFACT UNOA syntax, for example, this would be the question mark. Select `(none)` if this format version does not use the release character.
Segment tag separator	This field is automatically set to default. Here you can select the character used for separating segments. For the EDIFACT UNOA syntax, for example, this would be the single quote, and in ANSI X-12 the carriage return (CR).
Segment terminator	This field is automatically set to default. Here you can select the character used for terminating segments.
Wrap segments	This field is automatically set to No. Here you have to indicate if segments that are too long should be wrapped. Some formats have restrictions for segment lengths. The way in which wrapping should take place can be defined below.

Supported document types

In this section you specify the document types supported by the send port.

Use this	To do this
Accept all unlisted documents	Specify whether to accept unlisted documents.
Default EDIFACT format version	Specify the default EDIFACT format version.
Default X12 format version	Specify the default X12 format version.

Supported document types for EDIFACT

This is a required field. Select from the list of supported EDIFACT document types. You are authorizing this send port to accept only certain EDIFACT document formats.

Supported document types for X-12

This is a required field. Select from the list of supported X-12 document types. You are authorizing this send port to accept only certain X-12 document formats.

4. Click **OK** and **OK** again to save settings.

Developing EDI Business Processes

This section provides procedural and conceptual information describing the steps to develop an EDI business process. The focus lies on illustrating the function of each component and the interconnections of each aspect within the overall EDI business process.

Before developing an EDI business process, ensure that you are part of the EDI subsystem user group. Contact your system administrator for details.

In This Section

- Developing Parties for EDI Business Processes
- Developing Party Aliases for EDI Business Processes
- Developing Send Ports for EDI Business Processes
- Developing Receive Locations for EDI Business Processes

Developing Parties for EDI Business Processes

A party represents a trading partner with whom you have agreed to electronically exchange documents in a certain format. A party is the start and end point of the EDI business process, because a party represents the initial sender of the document along with the final recipient.

Each party must have a unique characteristic to identify it, a party alias.

Developing Party Aliases for EDI Business Processes

Each party is identified by a unique party alias, which consists of three distinct elements: name, qualifier, and value. Each party must have at least one set of party aliases.

Use this	To do this
Name	If a party uses a mutually agreed-upon EDI value to qualify itself, the **Name** of the qualifier is EDI. For example, if a telephone number is used to identify the party, then the entry in the **Name** field would be the telephone number. When developing an EDI solution, the **Name** value is EDI.
Qualifier	The entry in the **Qualifier** field identifies the entry in the **Value** field. For example, if you enter EDI in the **Qualifier** field, you are classifying the entry in the **Value** field as an EDI value. When developing an EDI solution, the **Qualifier** value must be EDI.
Value	The entry in the **Value** field is the mutually agreed-upon EDI logical address or telephone number. For EDI, the value must adhere to the following structure: EDI://aaaa:bb:zzzzz

How the Base EDI Adapter Uses Party Aliases

1. A party alias is required to determine the recipient for inbound messages.

2. The Base EDI adapter takes the logical address of the message sender, creates a URI from this logical address, and then finds the receive location record with that URI.

3. Next, the Base EDI adapter takes the logical address of the message recipient, creates a URI from that and using this URI, finds the party alias record.

4. Finally, the Base EDI adapter creates the message context from the sender URI and the recipient qualifier and value. BizTalk Server then routes the message to the recipient.

Developing Send Ports for EDI Business Processes

Send ports are used to send messages to trading parties. When configuring the send port, you are determining the specifics of sending a message, such as the transport type, retry count, and retry interval to be used. The send port is configured with details relating to the recipient of the interchange.

When EDI is used as the transport type for a send port, you can specify that the send port only accepts certain EDI document formats, along with the folder from which translated documents can be retrieved.

The value in the retry count field determines the number of times the Base EDI adapter attempts to resend a document.

The value of the retry interval property determines the amount of time BizTalk Server waits before it attempts to resend the message. The default value of the retry interval property is five minutes.

The send port can be incorporated into an EDI business process by linking the send port to a party and/or receive location.

By linking a send port to a receive location, documents sent through this receive location are sent directly to the linked send port, which in turn can be associated with a particular party.

To link a send port to a receive location

1. On the **View** menu, click **BizTalk Explorer**.

2. Click the **Filters and Maps** section.

3. Click the **Filters** section.

4. In the **Property** field, select **BTS.ReceivePortName** from the drop-down list.

5. In the **Value** field, type the relevant receive location.

6. Click **OK**.

7. Right-click the send port, and then click **Enlist**.

You can link an associate party to a particular send port so that documents received at the specified send port are directly routed to the intended recipient.

To associate parties to send ports

1. On the **View** menu, click **BizTalk Explorer**.
2. Click the **Parties** folder.
3. Select a party from the available list.
4. Right-click the party, and then click **Edit**.
5. Click the **Send Ports** tab.
6. Select a send port from the drop-down list.
7. Click **OK**.

Developing Receive Locations for EDI Business Processes

A receive location is the channel through which you send files to a party. The receive location is the first step in the EDI process. When you configure the receive location, you determine the specifics of sending a message—for example, when EDI is used as the transport type, you can specify that the receive location only sends certain EDI document types, or you can specify the folder from which documents are picked up.

You link a receive location into the overall EDI business process by linking the receive location to a specific send port.

Developing Schemas and Maps for the Base EDI Adapter

Using the Microsoft BizTalk Server 2006 Base EDI adapter, you can customize individual EDI schemas to meet your business needs and those of your trading partners.

Ensure that you are part of the EDI subsystem user group before developing EDI schemas and maps. Contact your system administrator for details.

The schema editor you use depends on the type of modifications you want to make to a schema.

In This Section

- How to Enable the EDI Schema Editor Extension
- How to Validate a Schema

- How to Generate an Instance
- How to Validate an Instance

How to Enable the EDI Schema Editor Extension

BizTalk Editor is used to modify an individual XSD schema to your specific business needs and those of your trading partners.

The Covast EDI Schema Editor Extension should be enabled for each XSD schema you want to customize to modify EDI-specific properties.

To enable the Covast EDI Schema Editor Extension

1. Open the schema.
2. Select the <*Schema*> node.
3. Press F4 to open the Properties window.
4. Select **Schema Editor Extensions** and click the ellipsis (**...**) button.
5. Select **Covast EDI Schema Editor Extension**.
6. Click **OK**.

The document definition of the XSD schema is stored within several annotations throughout the XSD.

When you click the **EDI Schema** tab, you can see the complete document definition.

To view an EDI document definition schema

1. Click **Covast EDI Schema Editor Extension**.
2. Click the **EDI Schema** tab.

How to Validate a Schema

You validate an XSD schema to create an EDI representation of that schema and store it in the Base EDI adapter database.

Ensure that the environment variable ESP_SRV is set to <*drive*>:\Program Files\Microsoft BizTalk Server 2006\EDI\Subsystem. If the variable is not set to the correct installation location, an error may occur when validating a schema.

To validate an XSD Schema

1. Right-click the XSD schema.

2. Select **Validate Schema** from the drop-down list.

This process takes a few moments to complete.

How to Generate an Instance

Using the generate instance feature, you can create an XML test document that can be discarded.

To generate a test instance

1. Right-click the **XSD** schema.
2. Select **Generate Instance** from the drop-down list.

This process takes a few moments to complete.

How to Validate an Instance

Using the validate instance feature, you can syntactically compare an XML document to an XSD schema.

To validate an XSD instance, the following requirements must be met:

- The logical addresses of the sender and recipient must be defined as parties with the BizTalk Server Administration console or with the BizTalk Explorer available in Visual Studio 2005.
- An alias needs to be defined that adheres to EDI requirements—namely, the Name field must contain the value EDI, the Qualifier field must contain the value EDI and Value must follow the EDI structure: EDI://aaaa:bb:zzzzz.
- The sender of the interchange must create a receive location with EDI transport type, and the URI must be equal to the logical address.

To validate an instance

1. Right-click the XSD schema.
2. Select **Validate Instance** from the drop-down list.

This process takes a few seconds to complete.

Using EDI Acknowledgements

The sender of an EDI document often requests a receipt of their transmission. This provides nonrepudiation—for example, a party cannot, at a later date, claim that they did not receive a purchase order or invoice.

The EDI acknowledgement either confirms that the Base EDI adapter successfully processed the document, or it details any formatting errors that caused a loss of data. An EDI acknowledgement can be sent to acknowledge the original interchange or functional groups within the interchange.

You can configure an EDI acknowledgement so that it is always sent automatically. Alternatively, you can specify that the EDI acknowledgement is only sent when the document has failed. An EDI acknowledgement is sent within 12 hours of the initial transmission.

An EDI acknowledgement itself is never acknowledged to prevent endless cycles of acknowledgements between sender and recipient.

In This Section

- How to Configure an EDI Acknowledgement
- 997 Functional Acknowledgements
- TA1 Interchange Acknowledgements

How to Configure an EDI Acknowledgement

The BizTalk Server 2006 Base EDI adapter can be configured so that for each EDI document the Base EDI adapter sends to a trading partner, a functional acknowledgement is expected.

The length of time that can elapse before the Base EDI adapter expects to receive a functional acknowledgement is configured in the Time-outs section of the BizTalk Server 2006 Base EDI Administration console.

To configure the Base EDI adapter to request an EDI acknowledgement

1. In the BizTalk Server Administration console, right-click a send port, and then click **Properties**.

   –Or–

   In BizTalk Explorer, right-click a send port, and then click **Edit**.

2. Ensure that **EDI** is selected as the **Transport** type.

3. In the BizTalk Server Administration console, click **Configure**.

   –Or–

   In BizTalk Explorer, in the logical address field, click the ellipsis (**...**) button.

   This displays the **Transport Properties** dialog box.

4. Expand the **Flags** tree view.

5. Click **Functional Acknowledgement**.

Use this	To do this
Select **Always**	Selecting **Always** means that the Base EDI adapter expects a functional acknowledgement regardless of whether the document is accepted or rejected.
Select **Never**	Selecting **Never** means that the Base EDI adapter expects a functional acknowledgement only when the document is rejected.

You can also configure the Base EDI adapter so that a functional acknowledgement is sent for each EDI document received by the Base EDI adapter from a trading partner.

To configure the Base EDI adapter to send an EDI acknowledgement

1. In the BizTalk Server Administration console, right-click a receive location and then click **Properties**.

   –Or–

   In BizTalk Explorer, right-click a receive location, and then click **Edit**.

2. Ensure that **EDI** is selected as the **Transport** type.

3. In the BizTalk Server Administration console, click **Configure**.

   –Or–

   In BizTalk Explorer, in the logical address field, click the ellipsis (**...**) button,

   This displays the **Transport Properties** dialog box.

4. Expand the **Flags** tree view.

5. Click **Functional Acknowledgement**.

Use this	To do this
Select **Always**	Setting **Always** means that a functional acknowledgement is sent regardless of whether the Base EDI adapter accepts the document.
Select **Never**	Selecting **Never** means that a functional acknowledgement is sent only when the Base EDI adapter rejects the document.

997 Functional Acknowledgements

The 997 functional acknowledgement indicates the results of analysis of the syntax of the functional groups and documents within an X-12 interchange. A functional acknowledgement enables the sender to reformat the document according to the correct syntax, and then resubmit.

A single functional acknowledgement is sent for each acknowledged functional group.

997 Document Structure

Each segment within a 997 functional acknowledgement plays a specific role within the document. For example, the AK1 segment starts the acknowledgement of a functional group.

Segments are composed of a sequence of elements, which are the lowest level of information within a document. For example, invoice number can be an element.

In This Section

- ST Segment
- AK1 Segment
- AK2 Segment
- AK3 Segment
- AK4 Segment
- AK5 Segment
- AK9 Segment
- SE Segment

ST Segment

The ST segment acts as the transaction set header segment of an X-12 document (transaction set).

The ST segment is used to indicate the start of a transaction set within an interchange and is composed of the following mandatory elements.

ST Segment Elements

The elements in the ST segment specify the identifier code for the transaction set and the identifying control number.

Element	Purpose
Transaction Set Identifier Code	This element contains the identifier code for the transaction set, for example, 850.
Transaction Set Control Number	This element contains the identifying control number assigned by the sender of the transaction set.

AK1 Segment

The AK1 segment is mandatory within a 997 acknowledgement. The purpose of this segment is to start the acknowledgement of a functional group.

The AKI segment consists of the following mandatory elements.

AK1 Segment Elements

The elements in the AK1 segment provide information about the functional groups being acknowledged.

Element	Purpose
Functional Identifier Code	This code identifies a group of application-related transaction sets.
Group Control Number	The original sender of the transaction sets assigns this control number; this number refers to the functional groups that are acknowledged.

AK2 Segment

The AK2 segment is an optional segment within the 997 response file. The function of the AK2 segment is to indicate the start of a single transaction set.

The AK2 segment contains the following mandatory elements.

AK2 Segment Elements

The elements in the AK2 segment identify the transaction set.

Element	Purpose
Transaction Set Identifier Code	This code is assigned by the sender and acts to uniquely identify the transaction set.
Transaction Set Control Number	This control number, assigned by the original sender of the data, is unique within the transactional set functional group.

AK3 Segment

The AK3 segment is an optional segment within a 997 functional acknowledgement. The function of this segment is to report errors within a data segment and the location of that data segment.

The AK3 segment contains the following elements.

AK3 Segment Elements

The elements in the AK3 segment identify the location of the segment containing syntactical errors.

Element	Purpose
Segment ID Code	This code identifies the code ID of the data segment containing the error.
Segment Position in Transaction Set	This element provides the location of the segment containing syntactical errors from the start of the transaction set.
Segment Syntax Error Code	This optional element indicates the syntactical error found in the segment.

AK4 Segment

The AK4 segment is an optional segment within a 997 functional acknowledgement. The function of the AK4 segment is to provide details about erroneous data elements and report the location of those elements.

The AK4 segment consists of the following elements.

AK4 Segment Elements

The elements in the AK4 segment provide information about the syntax error and the position of the erroneous element in the segment.

Element	Purpose
Element Position in Segment	This mandatory element indicates the relevant position of the erroneous data element within a data segment.
Data Element Reference Number	You use this optional element to find the data element in the data element dictionary.
Data Element Syntax Error Code	This code indicates errors found after the element has been syntactically edited.

	1= Mandatory element missing 2= Conditional required data element missing 3= Too many data elements 4= Data element too short 5= Data element too long 6= Invalid character in data element 7= Invalid code value
Copy of Bad Element	This is a copy of the erroneous data element

AK5 Segment

The AK5 segment is mandatory for the 997 response message. The function of this segment is to acknowledge acceptance or rejection of the transaction set and to report errors within the transaction set.

This segment contains the following elements.

AK5 Segment Elements

The elements in the AK5 segment provide information about whether the transaction set is to be accepted or rejected due to syntactical errors.

Element	Purpose
Transaction Set Acknowledgement Code	This code indicates whether to accept or reject the transaction set based on the syntactical editing.
Transaction Set Syntax Error Code	This code, in this optional element, indicates the error found after the syntactical editing of a transaction set. 1= Transaction set not supported 2= Transaction set trailer missing 3= Transaction set control number in header and trailer do not match 4= Number of included segments does not match the actual count 5= One or more segments are in error

AK9 Segment

The AK9 segment is a mandatory segment within a 997 functional acknowledgement. This segment is used to acknowledge the acceptance or rejection of a functional group and report the number of transaction sets from the original trailer, the accepted sets, and the received sets within this functional group.

The AK9 segment contains the following elements.

AK9 Segment Elements

The elements in the AK9 segment provide information about the number of documents (transaction sets) included in the functional group and how many of these documents (transaction sets) were accepted by the Base EDI adapter.

Element	Purpose
Functional Group Acknowledgement Code	This code indicates whether to accept or reject the functional group based on the syntactical analysis. A= Accepted E= Accepted but the file contained errors P= Partially accepted, but a at least 1 transaction set was rejected

	R= Rejected
Number of Transaction Sets Included	This is the total number of transaction sets containing this data element.
Number of Received Transaction Sets	Number of transaction sets received
Number of Accepted Transaction Sets	Number of accepted transaction sets within a functional group.

SE Segment

The SE segment indicates the end of an X-12 document (transaction set). The SE segment contains the following mandatory elements.

SE Segment Elements

The SE segment elements provide information about the number of elements within a transaction set and the control number for that particular transaction set.

Element	Purpose
Number of Included Segments	This element checks the number of elements within a transaction set.
Transaction Set Control Number	This element checks the control number for the transaction set

TA1 Interchange Acknowledgements

The TA1 interchange acknowledgement is used to verify the syntactical correctness of the envelope of the X-12 interchange.

The TA1 interchange indicates that the file has been successfully received, and what errors existed within the envelope segments of the received X-12 file.

A TA1 response message always contains the ISA and IEA segments. If the error occurs in the functional group header or trailer (GS/GS) segments of the received X-12 file, however, the TA1 response also contains these elements.

TA1 Document Structure

The structure of a TAI interchange acknowledgement depends on the structure of the envelope of the original EDI document. When the envelope of the EDI document does not contain an error, then the interchange acknowledgement contains only the ISA, TAI, and IEA segments.

If the EDI document contains an error at the interchange level, such as in the interchange control header (ISA) segment or the interchange control trailer (IEA), then the interchange acknowledgement contains only the ISA, TA1, and IEA segments.

However, if the error occurs in the functional group header or trailer, then the interchange acknowledgement message also contains the GS, S, AK1, AK9, SE, and GE segments.

In This Section

- ISA Segments
- TA1 Segments
- GS Segments
- ST Segments
- AK1 Segments
- AK9 Segments
- SE Segments
- GE Segments
- IEA Segments

ISA Segments

The ISA segment is the interchange header for the X-12 document, and contains sender and recipient information for a set of X-12 documents. The interchange header is the outermost layer of the EDI envelope, and is always present in an interchange acknowledgement message. The following elements are contained within the ISA segment.

ISA Segment Elements

The elements within the ISA segment provide general information relating to the interchange, such as sender and recipient and the date and time the interchange was prepared. All elements within this segment are mandatory.

Element	Purpose
Authorization Information Qualifier	This is a code to determine the type of information in the authorization information element. This qualifier has a maximum length of two positions. The default value for this field is 00, which indicates that no authorization information is present.
Authorization Information	This element contains additional information about either the sender of the interchange or the data in the interchange. The content of this

	element is set by the Authorization Information Qualifier element.
Security Information Qualifier	This is a code to determine the type of information in the security information element. This qualifier has a maximum length of two positions. The default value for this field is 00, which indicates that no security information is present.
Security Information	This element identifies security information concerning the sender of the data in the interchange. The content of this element is set by the Security Information Qualifier element.
Interchange ID Qualifier	This element allocates the method of code structure that is used to designate the sender or receiver ID qualifier. The Interchange ID Qualifier defines the code used in the Interchange Sender element.
Interchange Sender	This element identifies the sender's ID. The ID code used by the sender is defined in the Interchange ID Qualifier.
Interchange Receiver	This element identifies the recipient's ID. The ID code used by the sender is defined in the Interchange ID Qualifier.
Date	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Time	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Interchange Standards Identifier	This element identifies the agency responsible for the control standard the message uses.
Interchange Version ID	This version number only applies to the interchange control segment. This version number only applies to the envelope and is not the same as the version number used in the GS segment.
Interchange Control Number	This number uniquely identifies the interchange. The sender assigns the interchange control number, and together with the sender ID uniquely identifies the interchange data to the recipient.
Acknowledgement Requested	The sender of the interchange determines this code; it is used to request an interchange acknowledgement. A zero value in this position indicates that an acknowledgement has not been requested.
Test Indicator	This code identifies whether the data contained within this interchange is for test or production purposes. A P value in this position indicates that the data is used for production purposes. A T value indicates that the data is used for test purposes.
Sub-Element Separator	This field is reserved for future expansion. This element is used for separating data element subgroups.

TA1 Segments

The TA1 segment is an optional segment. The function of this segment is to acknowledge the receipt of and/or the syntactical correctness of the envelope segments of an X-12 interchange.

The TA1 segment contains the following elements.

TA1 Segment Elements

The TA1 segment elements provide information about whether the interchange was accepted, and if an error occurred, what the nature of the error is.

Element	Purpose
Interchange Control Number	This number uniquely identifies the interchange. The sender assigns the interchange control number and this together with the sender ID uniquely identifies the interchange data to the recipient.
Interchange Date	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Interchange Time	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Interchange Acknowledgement Code	This is mutually agreed between the trading partners such as: • A = Accepted • R= Rejected • E= Accepted, but the file contains errors and must be resubmitted
Interchange Note Code	This is a three-digit number that corresponds to one of the following note codes: • 000 No errors • 001 The Interchange Control Number in the header and trailer do not match. The value in the header is used as an acknowledgement. • 002 The standard as noted in the Control Standards Identifier is not supported. • 003 The version of the controls is not supported. • 004 The segment terminator is not valid.

	• 005 Invalid interchange ID qualifier for sender • 006 Invalid interchange ID for sender • 007 Invalid interchange ID qualifier for recipient • 008 Invalid interchange ID for recipient • 009 Unknown interchange receiver ID • 010 Invalid Authorization Information Qualifier value • 011 Invalid Authorization Information value • 012 Invalid Security Information Qualifier value • 013 Invalid Security Information value • 014 Invalid Interchange Date value • 015 Invalid Interchange Time value • 016 Invalid Interchange Standards ID value • 017 Invalid Interchange Version ID number • 018 Invalid Interchange Control number • 019 Invalid Acknowledgment Request value • 020 Invalid Test Indicator value • 021 Invalid Number of Included Group value • 022 Invalid control structure • 023 Improper end of file • 024 Invalid Interchange content • 025 Duplicate Interchange Control number • 026 Invalid Data Element Separator • 027 Invalid Component Element Separator • 028 Invalid delivery date in the Deferred Delivery Request

	• 029 Invalid delivery time in the Deferred Delivery Request • 030 Invalid delivery time code in the Deferred Delivery Request • 031 Invalid grade of service code

GS Segments

This is the functional group header segment of a set of X-12 documents (transaction sets) of the same document type. The GS segment contains the following mandatory elements. The GS segment is only present in an interchange acknowledgement if an error occurs in the functional group header or trailer of the EDI document.

GS Segment Elements

The elements in this segment give information relating to the functional group, such as the codes identifying the sender and recipient, and the date and time of preparation.

Element	Purpose
Functional ID Code	This is element that identifies one type of message within the functional group such as Functional Acknowledgement (FA) or a Purchase Order transaction (PO).
Application Senders Code	This is a unique code identifying the sender of the interchange. The parties in the interchange must mutually agree this code.
Application Receivers Code	This is a unique code identifying the recipient of the interchange.
Date of Preparation	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Time of Preparation	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Group Control Number	The sender of the interchange maintains this number, which must be unique to each trading partner within an interchange.
Responsible Agency Code	This element is used in conjunction with the version element to identify the organization responsible the publication and maintenance of the message type.
Version/Release/Industry ID Code	This element states the version, release, and industry code of the functional group.

ST Segments

This is the Transaction Set Header segment of an X-12 document. The ST segment is used to indicate the start of a transaction set within an interchange The ST segment is only present in an interchange acknowledgement if an error occurs in the functional group header or trailer of the EDI document.

The ST segment contains the following mandatory elements.

ST Segment Elements

The elements contained in the ST segment provide information about the identifiers for the transaction set and the sender of the transaction set.

Element	Purpose
Transaction Set Identifier Code	This element contains the identifier code for the transaction set, that is, 850.
Transaction Set Control Number	This element contains the identifying control number assigned by the sender of the transaction set.

AK1 Segments

The AK1 segment is mandatory within a 997 acknowledgment. The purpose of this segment is to start the acknowledgment of a functional group. The AK1 segment is only present in an interchange acknowledgement if an error occurs in the functional group header or trailer of the EDI document.

The AKI segment consists of the following mandatory elements.

AK1 Segment Elements

The elements in the AK1 segment provide both the functional identifier code and the group control numbers.

Element	Purpose
Functional Identifier Code	This code identifies a group of application-related transaction sets.
Group Control Number	The original sender of the transaction sets this control number; this number refers to the functional groups being acknowledged.

AK9 Segments

The AK9 segment is a mandatory segment within a 997 functional acknowledgement. This segment is used to acknowledge the acceptance or rejection of a functional group and to report the number of transaction sets from the original trailer, the accepted sets, and the received sets within this functional group.

The AK9 segment is only present in an interchange acknowledgement if an error occurs in the functional group header or trailer of the EDI document.

The AK9 segment contains the following elements.

AK9 Segment Elements

The elements in the AK9 segment provide information about the number of documents (transaction sets) included in the functional group and how many of these documents (transaction sets) were accepted by the Base EDI adapter.

Element	Purpose
Functional Group Acknowledgment Code	This code indicates whether to accept or reject the conditions based on the syntactical analysis of the functional group. A= Accepted E= Accepted, but the file contained errors P= Partially accepted, but at least one transaction set was rejected R= Rejected
Number of Transaction Sets Included	This is the total number of transaction sets containing this data element.
Number of Received Transaction Sets	Number of transaction sets received.
Number of Accepted Transaction Sets	Number of accepted transaction sets within a functional group.

SE Segments

The SE segment is the transaction set trailer segment of an X-12 transaction set. The SE segment is used to indicate the end of a transaction set. The SE segment is only present in an interchange acknowledgement if an error occurs in the functional group header or trailer of the EDI document.

The SE segment contains the following mandatory elements.

SE Segment Elements

The SE segment elements provide information about the number of elements within a transaction set and the control number for that particular transaction set.

Element	Purpose
Number of Included Segments	This element checks the number of elements within a transaction set.
Transaction Set Control number	This element checks the control number for the transaction set

GE Segments

The GE segment is the functional group trailer segment of a set of X-12 documents (transaction sets) of the same document type.

The function of the GE segment is to indicate the end of a group of transaction sets in an interchange

The GE segment contains the following mandatory elements.

GE Segment Elements

The GE segment gives information about the number of functional groups within an interchange and the group control number.

Element	Purpose
Number of Transaction Sets Included	This number, generated by the supplier, acts as a record of the total number of functional groups or transaction sets within an interchange.
Group Control Number	This number, generated by the supplier, must be the same as the data interchange number in the GS segment.

IEA Segments

This is the Interchange Trailer segment of a set of X-12 functional groups. The function of this segment is to indicate the end of an interchange.

This segment includes the following mandatory elements.

IEA Segment Elements

The elements in the IEA segment provide information about the number of functional groups included in the interchange and the interchange control reference number.

Element	Purpose
Number of Included Functional Groups	This number is a record of the number of functional groups within the interchange. The supplier generates this number.
Interchange Control Reference	This number is used to uniquely identify the interchange data to the sender. This number must be the same as the interchange control number within the ISA segment.

Working with Custom Reference Numbers

Custom reference numbers (CRN) are a distinctive sequence of numbers that can be assigned for each party, at the interchange, group, or document level.

In This Section

- Using Custom Reference Numbers
- How to Configure Custom Reference Numbers

Using Custom Reference Numbers

The key to accruing benefits from custom reference numbers is to ensure that the value in the **Next value** field, along with the document format and type, are mutually agreed upon between each partner beforehand.

For example, you determine with your trading partner that all purchase order documents you send will be in the X-12 4010 850 document format. You agree that the first purchase order will have 0001 in the header segment of the EDI envelope (ISA and GS in X-12). This is set in the Value field of custom reference numbers and the maximum value in the header field is 1000.

In this example, using the custom reference number helps to structure the EDI relationship with your trading partner. Each purchase order has a sequential number, which reduces the possibility of your trading partner processing the same purchase order twice, or not realizing that a purchase order has not been delivered. For example, suppose that the second-to-last 850 received had the value 0063 in the header segment, and the most recently received has 0065. Therefore, you can conclude that your trading partner has sent a purchase order that you have not received.

How to Configure Custom Reference Numbers

Configure custom reference numbers for an EDI send port by following these steps.

To configure custom reference numbers for an EDI send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. On the **General** tab, in the **Transport** section, for the **Type** option, specify **EDI**.

2. On the **General** tab, in the **Transport** section, next to **Type**, click **Configure**.

3. In the **EDI Transport Properties** dialog box click the ellipses button (…) to display the **EDI Address (URI) and Custom Reference Numbering properties** dialog box.

4. In the **EDI Address (URI) and Custom Reference Numbering properties** dialog box under **Custom Reference Numbering for Selected EDI Address (URI):** do the following:

Use this	To do this
Type	Select at which level the custom reference number will be applied. Select from Interchange, Group, or Document.
From Format	Select the document format to which the custom number reference number will be applied.
Document type	Select the document type to which the custom reference number will be applied.
Next value	Insert the next value for the custom reference number.
Minimum value	Determine the minimum value for this custom reference number.
Maximum value	Determine the maximum value for this custom reference number.

Using EDIFACT EDI Segments

An EDIFACT transmission or interchange consists of a number of mandatory or conditional segments, which constitute the "envelope" for the transmission as a whole and for messages and groups of messages within it. Each segment plays a specific role in an EDIFACT document. A segment is composed of a sequence of elements, which is the lowest level of information within a document. For example, Invoice number could be an element.

The structure of an EDIFACT document is organized into three levels: interchange level, group level, and message level.

An interchange begins with a UNA or UNB segment and ends with a UNZ segment. A group begins with a UNG segment and ends with a UNE segment. A message begins with a UNH segment and ends with a UNT segment.

All EDIFACT segments are named using a three-letter identifier, and end with a segment separator.

An EDIFACT document contains the following mandatory segments:

- UNA segment
- UNB segment
- UNG segment
- UNH segment
- UNT segment
- UNE segment
- UNZ Segment

This section contains an overview of the function of each segment within an EDIFACT document, along with the purpose of each element composing the segments.

In This Section

- UNA Segments
- UNB Segments
- UNG Segments
- UNH Segments
- UNT Segments
- UNE Segments
- UNZ Segments

UNA Segments

The UNA segment is optional within an EDIFACT interchange. The specifications in the UNA segment define the characters that are used as separators and indicators for the interchange.

The UNA segment is used only when the conditions specified in this segment surpass the specifications in the UNB segment.

This UNA segment is used only if the separators deviate from the standard ones.

The UNA segment contains the following mandatory elements.

UNA Segment Elements

The UNA segment elements provide information about the data separators used in the interchange.

Element	Purpose
Component Data Element Separator	This element acts as a composite element delimiter. The default character for this segment is a colon (:).
Data Element Separator	This element acts as a data element delimiter. The default character for this segment is a plus sign (+).
Decimal Notification	The recipient ignores the character transferred in this position. The default character for this segment is a full stop/period (.). This segment is maintained to ensure upward compatibility with earlier versions of the syntax.
Release Indicator	This character is used to indicate that the text following contains one of the characters used as a composite, data, or segment separator. Therefore, this character is released from its conventional usage in this instance. The default character for this segment is a question mark (?)
Reserved for future use	This segment inserts a space where all valid standard codes are used. The default character for this segment is a blank.
Segment Terminator	This segment is used to indicate the end of the current segment and the start of a new segment. The default character for this segment is an apostrophe (').

UNB Segments

The UNB segment is mandatory to an EDIFACT interchange. This segment acts as the interchange header for a set of EDIFACT documents.

The UNB segment contains the following elements.

UNB Segment Elements

The UNB segment elements identify the sender and recipient of the interchange along with the date and time that the interchange was prepared and the agency controlling the syntax of the interchange.

Element	Purpose
Syntax Identifier	This mandatory element identifies both the level of the syntax and the agency controlling the syntax, that is UNOA.
Syntax Version Number	This mandatory element indicates the version number identified in the Syntax Identifier.
Sender Identification	This mandatory element indicates the name of or a code for the sender of the data.
Identification Code Qualifier	The Identification Code Qualifier is an optional element, referring to the sources for the codes for the identifiers of the interchanging partners, that is, zz indicates that the codes have been mutually defined.
Address for Reverse Routing	This optional element is the address the sender of the message specifies that the recipient is to include in response interchanges, to facilitate internal routing.
Recipient identification	This mandatory element indicates the name of or a code for the recipient of the data.
Identification Code Qualifier	The Identification Code Qualifier is an optional element, referring to the sources for the codes for the identifiers of the interchanging partners, that is, zz indicates that the codes have been mutually defined.
Routing Address	This optional element is the address the recipient of the message specifies that the sender is to include. The recipient uses this address for routing of received messages internally within the organization.
Date of Preparation	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Time of Preparation	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Interchange Control Reference	This mandatory element is a unique reference that the sender assigns to the interchange.
Recipient Reference Password	This optional element indicates the reference or password for the recipient that is mutually defined between the partners.
Recipient Reference Password/Qualifier	This optional element indicates the qualifier for the reference or password for the recipient.
Application Reference	This optional element identifies the application area assigned by the sender that the messages relate to.

Processing Priority Code	This is an optional element. This code indicates the processing priority for the interchange requested from the sender.
Acknowledgement Request	This optional element indicates the sender for the acknowledgement of the interchange.
Communications Agreement Id	This optional element indicates by name or code the type of agreement under which the interchange took place.
Test Indicator	This optional element indicates that the interchange is a test.

UNG Segments

The use of the UNG segment is optional. The function of this segment is to act as a header identifying and specifying a functional group.

The UNG segment contains the following elements.

UNG Segment Elements

The UNG segment elements contain information about the date and time that the functional group was prepared along with the type and version of the document in the functional group.

Element	Purpose
Functional Group Identification	This mandatory element identifies one type of message within the functional group.
Application Sender identification	This mandatory element indicates the department, division, etc. that the message was sent from.
Partner Identification Code Qualifier	The Identification Code Qualifier is an optional element. This element determines the codes for the identifiers of the interchanging partners, that is, ZZ means that they are mutually defined.
Application Recipient identification	This mandatory element indicates the name of, or a code for, the recipient of the data.
Partner Identification Code Qualifier	The Identification Code Qualifier is an optional element. This element determines the codes for the identifiers of the interchanging partners, that is, ZZ means that they are mutually defined.
Date of Preparation	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Time of Preparation	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Functional Group	This mandatory element is unique to the functional group. The sender of

Reference Number	the interchange assigns this number.
Controlling Agency	This is a mandatory element, which names the organization responsible for the publication and maintenance of the message type.
Message version	This mandatory element specifies the version of messages in the functional group.
Message type version number	This mandatory element indicates the version number of a message type.
Message type release number	This mandatory element indicates the release number within the current message type version number.
Association Assigned Code	This optional element is assigned by the organization responsible for the publication and maintenance of the current message type. This element further identifies the message.
Application Password	This is an optional element. The application password is for the section, group, and so on of the recipient.

UNH Segments

This is the Message Header segment of an EDIFACT document. The UNH segment identifies the message type and version and which agency is responsible for its maintenance.

The segment is used to indicate the start of a document within an interchange and indicates the type of document that is following.

UNH Segment Elements

The UNH segment elements provide information about the message type, and the agency responsible for maintaining the publication of the message type.

Element	Purpose
Message Reference Number	This number is a unique reference number for the message. It is unique to this functional group, and is assigned by the sender. This number must match the number in the UNT segment.
Message Type	This is a code that describes the type of message, for example, invoice. This code is assigned by the controlling agency.
Message Type Version Number	This element indicates the version number of a message type.
Message Type Release Number	This element indicates the release number within the current message type version number.

Controlling Agency	A code that identifies the agency controlling the specification, maintenance, and publication of the message type.
Assigned Code	This code is assigned by the agency responsible for controlling the specification, maintenance, and publication of the message type.
Common Access Reference	This reference serves as a key to relate all subsequent transfers of data to the same business case or file.
Sequence of Transfer	This element details the sequence of messages.
First and Last Transfer	This element provides information about the first and last messages transmitted.

UNT Segments

This is the Message Trailer segment of an EDIFACT document. It includes the document reference and number of segments in the document.

The segment is used to indicate the end of a document and to check the document reference and number of segments in the document.

UNT Segment Elements

The UNT segment elements provide information about the number of segments in a message and the reference number for the message.

Element	Purpose
Number of Segments in the message	This number records the total number of segments in the message.
Message Reference Number	This number is a unique reference number for the message. The sender assigns this number, and it is unique to the sender.

UNE Segments

This segment is the functional group trailer. The function of this segment is to act as an endpoint for a functional group, and to check its completeness.

The use of this segment is mutually agreed upon between parties.

The UNE segment contains the following mandatory elements.

UNE Segment Elements

The UNE segment elements record the total number of messages in the functional group, and the unique reference number given to this functional group.

Element	Purpose
Number of Messages	This element acts as a record of the total number of messages in the functional group.
Functional Group Reference Number	This number is a unique reference number. The sender assigns it, and it is unique to this functional group.

UNZ Segments

This segment is the Interchange Trailer segment of an EDIFACT document and contains the following mandatory elements. The segment is used to indicate the end of an interchange and to check the interchange reference and number of documents in the interchange.

UNZ Segment Elements

The elements in this segment provide information about the interchange reference and the number of documents in the interchange.

Element	Purpose
Interchange Control Reference	This segment indicates the number of messages, or if used the number of functional groups, in an interchange.
Interchange Control Reference	This segment is a unique reference that the sender assigns to the interchange

Working with X-12 EDI Documents

X-12 is a messaging standard developed by the American National Standards Institute (ANSI). X-12 is mainly used in the United States.

EDI documents are structured using a combination of segments and elements.

- An X-12 EDI document contains the following mandatory segments:
  - ISA segment
  - GS segment
  - ST segment
  - SE segment

- GE segment
- IEA segment

Each segment plays a specific role in the document; for example, the ISA segment is the interchange header, while the IEA segment is the interchange trailer. Segments are composed of a sequence of elements, which are the lowest level of information within a document. For example, invoice number can be an element.

This section contains an overview of the purpose of each segment within an X-12 document, along with the elemental composition of each segment.

In This Section

- ISA Segment Elements
- GS Segment Elements
- ST Segment Elements
- SE Segment Elements
- GE Segment Elements
- IEA Segment Elements

ISA Segment Elements

The ISA segment is the interchange header for the X-12 document, and contains sender and recipient information for a set of X-12 documents. The interchange header is the outermost layer of the EDI envelope. The following elements are contained within the ISA segment.

ISA Segment Elements

The elements within the ISA segment provide general information about the interchange, such as sender and recipient and the date and time the interchange was prepared. All elements within this segment are mandatory.

Element	Purpose
Authorization Information Qualifier	This is a code to determine the type of information in the authorization information element. This qualifier has a maximum length of two positions. The default value for this field is 00, which indicates that no authorization information is present.
Authorization Information	This element contains additional information about either the sender of the interchange or the data in the interchange. The content of this element is set by the Authorization Information Qualifier element.
Security Information	This is a code to determine the type of information in the security

Qualifier	information element. This qualifier has a maximum length of two positions. The default value for this field is 00, which indicates that no security information is present.
Security Information	This element identifies security information about the sender of the data in the interchange. The content of this element is set by the Security Information Qualifier element.
Interchange ID Qualifier	This element allocates the method of code structure that will be used to designate the sender or receiver ID qualifier. The interchange ID qualifier defines the code used in the interchange sender element.
Interchange Sender	This element identifies the sender ID. The ID code used by the sender is defined in the interchange ID qualifier.
Interchange Receiver	This element identifies the recipient ID. The ID code used by the sender is defined in the interchange ID qualifier.
Date	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Time	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Interchange Standards Identifier	This element identifies the agency responsible for the control standard used by the message.
Interchange Version ID	This version number only applies to the interchange control segment. This version number only applies to the envelope and is not the same as the version number used in the GS segment.
Interchange Control Number	This number uniquely identifies the interchange. The sender assigns the interchange control number and together with the sender ID uniquely identifies the interchange data to the recipient.
Acknowledgement Requested	The sender of the interchange determines this code as it is used to request an interchange acknowledgement. A zero value in this position indicates that an acknowledgement has not been requested.
Test Indicator	This code identifies whether the data contained within this interchange is for test or production purposes. A P value in this position indicates that the data is used for production purposes. A T value denotes that the data is used for test purposes.
Sub-Element Separator	This field is reserved for future expansion. This element is used for separating data element subgroups.

GS Segment Elements

This is the functional group header segment of a set of X-12 documents (transaction sets) of the same document type. The GS segment contains the following mandatory elements.

GS Segment Elements

The elements in this segment give information about the functional group, such as the codes identifying the sender and recipient, and the date and time of preparation.

Element	Purpose
Functional ID Code	This is element that identifies one type of message within the functional group, such as Functional Acknowledgement (FA) or a Purchase Order transaction (PO).
Application Senders Code	This is a unique code identifying the sender of the interchange. The parties in the interchange must mutually agree to this code.
Application Receivers Code	This is a unique code identifying the recipient of the interchange.
Date of Preparation	This mandatory element indicates the date that the interchange was prepared. This information is presented in the YYMMDD format.
Time of Preparation	This mandatory element indicates the time that the interchange was prepared. This information is presented in the 24-hour clock format.
Group Control Number	The sender of the interchange maintains this number, which must be unique to each trading partner within an interchange.
Responsible Agency Code	This element is used with the version element to identify the organization responsible for the publication and maintenance of the message type.
Version/Release/Industry id code	This element states the version, release, and industry code of the functional group.

ST Segment Elements

This is the transaction set header segment of an X-12 document. The ST segment is used to indicate the start of a transaction set within an interchange.

The ST segment contains the following mandatory elements.

ST Segment Elements

The elements contained in the ST segment provide information about the identifiers for the transaction set and the sender of the transaction set.

Element	Purpose
Transaction Set Identifier Code	This element contains the identifier code for the transaction set, for example, 850.
Transaction Set Control Number	This element contains the identifying control number assigned by the sender of the transaction set.

SE Segment Elements

The SE segment is the transaction set trailer segment of an X-12 transaction set. The SE segment is used to indicate the end of a transaction set.

The SE segment contains the following mandatory elements.

SE Segment Elements

The SE segment elements provide information about the number of elements within a transaction set and the control number for that transaction set.

Element	Purpose
Number of Included Segments	This element checks the number of elements within a transaction set.
Transaction Set Control Number	This element checks the control number for the transaction set

GE Segment Elements

The GE segment is the functional group trailer segment of a set of X-12 documents (transaction sets) of the same document type.

The function of the GE segment is to indicate the end of a group of transaction sets in an interchange

The GE segment contains the following mandatory elements.

GE Segment Elements

The GE segment gives information about the number of functional groups within an interchange and the group control number.

Element	Purpose
Number of Transaction Sets Included	This number, generated by the supplier, acts as a record of the total number of functional groups or transaction sets within an interchange.
Group Control Number	This number, generated by the supplier, must be the same as the data interchange number in the GS segment.

IEA Segment Elements

This is the interchange trailer segment of a set of X-12 functional groups. The function of this segment is to indicate the end of an interchange.

The IEA segment includes the following mandatory elements.

IEA Segment Elements

The elements in the IEA segment provide information about the number of functional groups included in the interchange and the interchange control reference number.

Element	Purpose
Number of Included Functional Groups	This number is a record of the number of functional groups within the interchange. The supplier generates this number.
Interchange Control Reference	This number is used to uniquely identify the interchange data to the sender. This number must be the same as the interchange control number within the ISA segment.

Working with EDI Annotations

An XSD schema can contain several annotations that are specific to EDI This topic describes how to view EDI annotations and enumerates the list of EDI annotations that can be found in an XSD schema.

In This Section

- How to View EDI Annotations
- Possible EDI Annotations

How to View EDI Annotations

EDI annotations can be added throughout the entire XSD schema. You can view all the annotations throughout a particular schema.

To view all EDI annotations within a schema

1. Open an XSD schema.
2. Click the **EDI Schema** tab.

This displays all the EDI annotations within the schema.

Possible EDI Annotations

The table in this section contains a list of EDI annotations that can be found in an XSD schema, along with a brief explanation of the annotation.

Annotation	Description
Standards Version	This annotation indicates the version of the document standard being used.
Subdocument Break	This annotation is used for supporting subdocument breaking. That is where, for example, an 837 transaction contains multiple claims, and the parser can break out individual work items for each Summing Claim (CLM) loop rather than treat the whole document as one work item. This property is set at the schema level to indicate that, somewhere in the document, there exists a breakpoint (something you typically want to know in advance).
Tag Name	This annotation indicates the tag name at the start of the segment, for example BEG or UNH.
Structure	This annotation indicates a non-XML structure indication, which can be either delimited or positional.
Count Ignore	This annotation indicates whether a node in the schema counts against the total for determining whether the number of segments matches the footer totals (in GE or UNT). This is necessary because loop container nodes exist in the schema, but do not physically exist in the data.
Trigger Field	Indicates what the trigger field for a particular record might be. For example, suppose that a schema contains two peer segments called N1BillTo and N1ShipTo. Their tags are the same, and the only way to tell the difference is by their qualifier in N101. BT indicates the first and ST indicates the second.
Trigger Value	See above.
EDI Datatype	This annotation indicates the EDI data type. These values match the standard

	(AN, ID, TM, DT, R, N, Nx where x=1-9, Dx where x=1-4, and so on).
Justification	This is applicable for fields whose length is less than the minimum length so that padding (zero or space) occurs. Left-justified causes padding on the right, and vice versa.
Format	This is the format for time and date data types for X-12.
Custom maxLength	This annotation indicates the length after XML normalization has occurred. For example, if the X-12 data type is N5, the data comes in as "123456", but when converted to XML, it becomes "1.234567" and so the custom maxLength accounts for the EDI length. The reasoning is that, if you use maxLength of an XML schema to control, you get a validation failure that might not actually be a problem.
Subjects	In X-12, there are three types of syntax rules: any, grouped, and all.
Name	See above; these are the subjects of syntax rules.

Base EDI Adapter Supported Standards

This section contains information about standards supported by the Base EDI adapter. The Base EDI Adapter supports two basic subsets of the EDI standard, X-12 and EDIFACT. This topic enumerates the document types supported within each subset and also lists the EDIFACT CONTRL message codes that are supported for use with the Base EDI adapter.

In This Section

- Supported X-12 Standards
- Supported EDIFACT Standards
- Supported EDIFACT CONTRL Message Codes

Supported X-12 Standards

The following table lists the X-12 standards that the Base EDI adapter supports.

X-12 standard	Used for
X-12-2040-810	Invoice
X-12-2040-832	Price/Sales Catalog
X-12-2040-846	Inventory Inquiry/Advice
X-12-2040-850	Purchase Order

X-12-2040-855	Purchase Order Acknowledgement
X-12-2040-856	Ship Notice/Manifest
X-12-2040-861	Receiving Advice/Acceptance Certificate
X-12-2040-864	Text Message
X-12-2040-867	Product Transfer and Resale Report
X-12-2040-997	Functional Acknowledgement
X-12-3010-810	Invoice
X-12-3010-832	Price/Sales Catalog
X-12-3010-846	Inventory Inquiry/Advice
X-12-3010-850	Purchase Order
X-12-3010-852	Product Activity Data
X-12-3010-855	Purchase Order Acknowledgement
X-12-3010-856	Ship Notice/Manifest
X-12-3010-861	Receiving Advice/Acceptance Certificate
X-12-3010-864	Text Message
X-12-3010-867	Product Transfer and Resale Report
X-12-3060-940	Warehouse Shipping Order
X-12-3060-944	Warehouse Stock Transfer Receipt Advice
X-12-3060-997	Functional Acknowledgement
X-12-3060-810	Ship Notice/Manifest
X-12-3060-832	Price/Sales Catalog
X-12-3060-846	Inventory Inquiry/Advice
X-12-3060-850	Purchase Order
X-12-3060-852	Product Activity Data

X-12-3060-855	Purchase Order Acknowledgement
X-12-3060-856	Ship Notice/Manifest
X-12-3060-861	Receiving Advice/Acceptance Certificate
X-12-3060-864	Text Message
X-12-3060-867	Product Transfer and Resale Report
X-12-3060-940	Warehouse Shipping Order
X-12-3060-944	Warehouse Stock Transfer Receipt Advice
X-12-3060-997	Functional Acknowledgement
X-12-4010-810	Invoice
X-12-4010-832	Price/Sales Catalog
X-12-4010-846	Inventory Inquiry/Advice
X-12-4010-850	Purchase Order
X-12-4010-852	Product Activity Data
X-12-4010-855	Purchase Order Acknowledgement
X-12-4010-856	Ship Notice/Manifest
X-12-4010-861	Receiving Advice/Acceptance Certificate
X-12-4010-864	Text Message
X-12-4010-867	Product Transfer and Resale Report
X-12-4010-940	Warehouse Shipping Order
X-12-4010-944	Warehouse Stock Transfer Receipt Advice
X-12-4010-997	Functional Acknowledgement

Supported EDIFACT Standards

The United Nations releases several new editions of the EDIFACT standards each year. The following table lists the EDIFACT standards that the Base EDI adapter supports.

EDIFACT standard	Used for
D93A-DESADV	Dispatch Advice
D93A-INVOIC	Invoice
D93A-INVRPT	Inventory Report
D93A-ORDERS	Purchase Order
D93A-ORDRSP	Order Response
D93A-PARTIN	Party Information
D93A-PAYEXT	Extended Payment Order
D93A- PRICAT	Price/Sales Catalogue
D93A- SLSRPT	Sales Data Report
D95A-APERAK	Application Error and Acknowledgement
D95A-DESADV	Dispatch Advice
D95A-INVOIC	Invoice
D95A-INVRPT	Inventory Report
D95A-ORDERS	Purchase Order
D95A-ORDRSP	Order Response
D95A-PARTIN	Party Information
D95A-PAYEXT	Extended Payment Order
D95A-PRICAT	Price/Sales Catalogue
D95A-SLSRPT	Sales Data Report
D95B-APERAK	Application Error and Acknowledgement
D95B-DESADV	Dispatch Advice

D95B-INVOIC	Invoice
D95B-INVRPT	Inventory Report
D95B-ORDERS	Purchase Order
D95B-ORDRSP	Order Response
D95B-PARTIN	Party Information
D95B-PAYEXT	Extended Payment Order
D95B-PRICAT	Price/Sales Catalogue
D95B-SLSRPT	Sales Data Report
D97B-APERAK	Application Error and Acknowledgement
D97B-DESADV	Dispatch Advice
D97B-INVOIC	Invoice
D97B-INVRPT	Inventory Report
D97B-ORDERS	Purchase Order
D97B-ORDRSP	Order Response
D97B-PARTIN	Party Information
D97B-PAYEXT	Extended Payment Order
D97B-PRICAT	Price/Sales Catalogue
D97B-PRODAT	Product Data
D97B-RECADV	Receiving Advice
D97B-SLSRPT	Sales Data Report
D98A-APERAK	Application Error and Acknowledgement
D98A-CONTRL	Syntax and Service Report
D98A-DESADV	Dispatch Advice
D98A-INVOIC	Invoice

D98A-INVRPT	Inventory Report
D98A-ORDERS	Purchase Order
D98A-ORDRSP	Order Response
D98A-PARTIN	Party Information
D98A-PAYEXT	Extended Payment Order
D98A-PRICAT	Price/Sales Catalogue
D98A-PRODAT	Product Data
D98A-RECADV	Receiving Advice
D98A-SLSRPT	Sales Data Report
D98B-APERAK	Application Error and Acknowledgement
D98B-CONTRL	Syntax and Service Report
D98B-DESADV	Dispatch Advice
D98B-INVOIC	Invoice
D98B-INVRPT	Inventory Report
D98B-ORDERS	Purchase Order
D98B-ORDRSP	Order Response
D98B-PARTIN	Party Information
D98B-PAYEXT	Extended Payment Order
D98B-PRICAT	Price/Sales Catalogue
D98B-PRODAT	Product Data
D98B-RECADV	Receiving Advice
D98B-SLSRPT	Sales Data Report

Supported EDIFACT CONTRL Message Codes

The following table lists the UN/EDIFACT Acknowledgement/Rejection Advice (CONTRL) error codes that the Base EDI adapter supports and provides a brief description of each.

Codes supported in received CONTRL messages	Description
Interchange Level (UCI) 2	Syntax version or level not supported The syntax version and/or level are not supported by the recipient.
Interchange Level (UCI) 7	Interchange recipient not actual recipient The interchange recipient (S003) is different from the actual recipient.
Interchange Level (UCI) 17	Invalid value The value of a simple data element, composite data element, or component data element does not conform to the specifications for the value.
Interchange Level (UCI) 23	Unknown Interchange sender The interchange sender (S002) is unknown.
Interchange Level (UCI) 26	Duplicate detected A possible duplication of a previously received interchange, functional group, or message has been detected.
Interchange Level (UCI) 28	References do not match The control reference in UNB/UNG/UNH does not match the one in UNZ/UNE/UNT.
Interchange Level (UCI) 29	Control count does not match number of instances received The number of functional groups/messages/segments does not match the number given in UNZ/UNE/UNT.
Interchange Level (UCI) 43	Unknown interchange recipient The recipient of the interchange could not be determined.
Functional Group	Missing segment or element

Level (UCF) 28	A mandatory (or otherwise required) service or user segment, data element, composite data element, or component data element is missing.
Functional Group Level (UCF) 29	Control count does not match number of instances received The number of functional groups does not match the number given in UNZ/UNE/UNT.
Functional Group Level (UCF) 30	Functional groups and messages mixed Individual messages and functional groups have been mixed at the same level in the interchange.
Functional Group Level (UCF) 31	More than one message type in group Different message types are contained within a functional group.
Functional Group Level (UCF) 32	Lower level empty The functional group did not contain any messages.
Functional Group Level (UCF) 33	Invalid occurrence outside message or functional group An invalid segment or data element occurred in the interchange, between messages or between functional groups.
Message Level (UCM) 30	Functional groups and messages mixed Individual messages and functional groups have been mixed at the same level in the interchange.
Message Level (UCM) 31	More than one message type in group Different message types are contained in a functional group.
Message Level (UCM) 32	Lower level empty The functional group did not contain any messages.
Message Level (UCM) 33	Invalid occurrence outside message or functional group An invalid segment or data element occurred in the interchange, between messages or between functional groups.
Segment Level (UCS) 13	Missing A mandatory (or otherwise required) service or user segment, data

	element, composite data element or component data element is missing.
Segment Level (UCS) 15	Not supported in this position The recipient does not support the use of the segment type, simple data element type, composite data element type, or component data element type in the identified position.
Segment Level (UCS) 16	Too many constituents The identified segment contained too many data elements or the identified composite data element contained too many component data elements.
Segment Level (UCS) 35	Too many segment repetitions A segment was repeated too many times.
Element Level (UCD) 12	Invalid value The value of a data element, composite data element, or component data element does not conform to the specifications for the value.
Element Level (UCD) 13	Missing A mandatory (or otherwise required) service or user segment, data element, composite data element, or component data element is missing.
Element Level (UCD) 14	Value not supported in this position The recipient does not support the use of the specific value of an identified simple data element, composite data element, or component data element in the position where it is used. The value may still be valid according to the specification if it is used in another position.
Element Level (UCD) 15	Value not supported in this position The recipient does not support use of the segment type, simple data element type, composite data element type, or component data element type in the identified position.
Element Level (UCD) 16	Too many constituents The identified segment contained too many data elements or the identified composite data element contained too many component data elements.
Element Level	Invalid decimal notation

(UCD) 19	The character indicated as decimal notation in UNA is invalid, or the decimal notation used in a data element is not consistent with the one indicated in UNA.
Element Level (UCD) 21	Invalid character(s) One or more character(s) used in the interchange is not a valid character as defined by the syntax level indicated in UNB. The invalid character is part of the referenced level, or followed immediately after the identified part of the interchange.
Element Level (UCD) 37	Invalid type of character(s) One or more numeric characters were used in an alphabetic (component) data element or one or more alphabetic characters were used in a numeric (component) data element.
Element Level (UCD) 38	Missing digit in front of decimal sign A decimal sign is not preceded by one or more digits.
Element Level (UCD) 39	Data element too long The length of the data element received exceeded the maximum length specified in the data element description.
Element Level (UCD) 40	Data element too short The length of the data element received is shorter than the minimum length specified in the data element description.

Codes supported in sent CONTRL messages	**Description**
Interchange Level (UCI) 12	Invalid value The value of a data element, composite data element, or component data element does not conform to the specifications for the value.
Document Level (UCD) 13	Missing segment or element A mandatory (or otherwise required) service or user segment, data element, composite data element, or component data element is missing.
Document Level (UCD) 17	No agreement There is no agreement that allows the receipt of an interchange,

	functional group, or message with the value of the identified simple data element, composite data element, or component data element.
Document Level (UCD) 23	Unknown Interchange sender The interchange sender (S002) is unknown.
Document Level (UCD) 26	Duplicate detected A possible duplication of a previously received interchange, functional group, or message has been detected.
Document Level (UCD) 42	Recipient unknown The recipient of the message could not be determined.
Segment Level (UCS) 15	Not supported in this position The recipient does not support the use of the segment type, simple data element type, composite data element type, or component data element type in the identified position.
Segment Level (UCS) 35	Too many segment repetitions A segment was repeated too many times.
Element Level (UCD) 12	Invalid value The value of a data element, composite data element, or component data element does not conform to the specifications for the value.
Element Level (UCD) 13	Missing A mandatory (or otherwise required) service or user segment, data element, composite data element, or component data element is missing.
Element Level (UCD) 14	Value not supported in this position The recipient does not support the use of the specific value of an identified simple data element, composite data element, or component data element in the position where it is used. The value may still be valid according to the specification if it is used in another position.
Element Level (UCD) 18	Unspecified error An error has been identified, but the nature of the error could not be determined or was not reported.

Element (UCD) 39	Data element too long Level The length of the data element received exceeded the maximum length specified in the data element description

Base EDI Adapter Security Recommendations

You can use the Base EDI adapter for loading, downloading, and translation processes that you need for communication with trading partners that use EDI. For more information about the Base EDI adapter, see Base EDI Adapter . The following guidelines are recommended for securing and deploying the Base EDI adapter in your environment:

- The person configuring the Base EDI adapter must be a member of the BizTalk Administrators group and of the BizTalk Base EDI Users group.
- The Base EDI adapter uses the local port number 11010 by default. If you have another resource that uses this port, select a different port for EDI in the adapter properties page. The Base EDI adapter does not use this port for communication with other computers, so you do not need to configure this port in the firewall.
- For a minimum-permission configuration, it is recommended to use the same service account to run both the Base EDI service (BizTalk EDI subsystem) and the BizTalk Host instance that runs the Base EDI adapter. You should specify this account when running the BizTalk Configuration Wizard.
- If the account for the Base EDI service is different from the account for the host instance running the Base EDI adapter, you must do the following to ensure the Base EDI adapter works as expected:
  - The BizTalk Host instance account where the Base EDI adapter is running must be a member of the EDI Users group.
  - The account for the Base EDI service or the BizTalk Base EDI Users group (by default named **EDI Subsystem Users**) must be a member of the BTS_Host_Users SQL Server role.
- If you install only the administrative tools, you must configure registry keys for the Base EDI send handler. For more information, see Base EDI Adapter Configuration Prerequisites .
- You must ensure that the service accounts for the host instances running the Base EDI adapter have read and write permissions to pick up messages from the file location for the EDI messages.

Troubleshooting the Base EDI Adapter

This section lists problems you may encounter when using the Base EDI adapter and their possible causes and resolutions.

Error encountered: ERROR (82), interchangenr nnnnn :A socket error has occurred

Problem: Receive the run-time error: ERROR (82), interchangenr <*interchange number*>: A socket error has occurred. Contact the system administrator. COM_SendMessage(): Can't make a connection to BizTalk for document #<*host name*>. Errorcode is -2074 ()Host=<*host name*>, Device=[(null)], Protocol=[tcp], Port=<*port number*> System message: (10061) No connection could be made because the target machine actively refused it.

Cause: The send port for the orchestration is not started.

Resolution: Start the send port.

Errors when starting the Base EDI Subsystem service

Problem: The following errors may be showing in the event log when the Base EDI Subsystem service is started automatically after a reboot:

1. database error

Event Type:Error

Event Source:EDI Subsystem

Event Category:BizTalk Server 2006

Event ID:24

Date:2/22/2005

Time:5:05:13 PM

User:N/A

Computer:PATH2

Description:

Error encountered: ERROR (44):

A database error occurred. Contact the system administrator.

ConnectToDatabase(): dbopen failed to connect to database server. ServerName=[path2]

2. account information error

Event Type:Error

Event Source:EDI Subsystem

Event Category:BizTalk Server 2006

Event ID:24

Date:2/22/2005

Time:5:05:13 PM

User:N/A

Computer:PATH2

Description:

Error encountered: ERROR (4) :

The account information has not been configured on the receive handler.

PRTNR3-Account record [MSDEFAULT][MS01] is missing for this connector, ConnectToDatabase(): dbopen failed to connect to database server. ServerName=[path2]

Cause: This is an intermittent timing issue; SQL Server is not available yet when the EDI Subsystem is starting.

Resolution: On a single-computer installation, this can be fixed by creating a dependency for the Base EDI Subsystem service on SQL Server.

File cannot be opened error

Problem

An error similar to the following is generated when processing files with the Base EDI adapter:

Error encountered: ERROR (3) :

The file cannot be opened. Make sure that the file exists.File [\\BPI2X64<*servername*>\EDIDocsHome\documents\system\external\inbox\11068.in], open_afile(\\<*servername*>\EDIDocsHome\documents\system\external\inbox\11068.in,w+b) : err:System message: (5) Access is denied

Cause

When the Base EDI Adapter is configured, a file share named **EDIDocsHome** is created to serve as the working directory for the EDI Subsystem that is running on each BizTalk Server in a BizTalk Server group. The location of this file share is configurable. By default the EDIDocsHome file share that is created maps to the <*drive*>**:\Documents and Settings\All Users\Application Data\Microsoft\Biztalk Server 2006\EDI\Subsystem** directory of the BizTalk server that the configuration program is running on. If files from a previous installation of the Base EDI adapter exist in this directory then an error occurs when the Base EDI Adapter attempts to create a new file in the directory with the same name.

Resolution

If you want to configure the Base EDI Adapter to use the same location for the EDIDocsHome share that a previous installation of the Base EDI Adapter used, follow these steps:

- Stop the BizTalk Base EDI Service on all BizTalk Servers in the BizTalk group.
- Navigate to the location to be used for the EDIDocsHome file share.
- Check to see if any files exist in the \Documents\System\Internal\Inbox or the \Documents\System\Internal\Outbox subdirectories of this file share.
- Move these files to another folder or, if you are running in a development environment, simply delete these files.

Creating Custom EDI Schemas

A limited list of standard UN-EDIFACT and ANSI-X12 schemas is provided with the Base EDI adapter. When this is not sufficient, you can create custom EDI schemas. This document explains the steps and requirements for creating such a custom schema.

After you have successfully created and validated your custom schema you can start using it. For the Base EDI adapter to accept messages for this custom schema, the authorization must be set appropriately on the relevant receive locations and send ports.

For EDI receive locations, do the following:

1. Navigate to the EDI Transport Properties window.
2. In the "Supported Document Types" category, set the "Accept all unlisted documents" option to Yes.

For EDI send ports, do the following:

1. Navigate to the EDI Transport Properties window.
2. In the "Supported Document Types" category:

a. Set the "Accept all unlisted documents" option to Yes.

b. Select the appropriate envelope for the "Default EDIFACT format version" option.

c. Select the appropriate envelope for the "Default X12 format version" option.

BizTalk Message Queuing (MSMQT) Adapter

The BizTalk Message Queuing (also known as MSMQT) adapter is an implementation of the Microsoft Message Queuing protocol that is wire-compatible with Microsoft Message Queuing (also known as MSMQ). The MSMQT adapter is an integral part of BizTalk Server and uses BizTalk Server administration tools. Messages received into the MSMQT adapter are received into a virtual representation of a Microsoft Message Queuing queue that is tied directly to the BizTalk Server message store.

In This Section

- What Is the BizTalk Message Queuing Adapter?
- How to Install Microsoft Message Queuing and BizTalk Message Queuing Side-by-Side
- Configuring the MSMQT Adapter
- MSMQT Adapter Deployment and Security Recommendations

What Is the BizTalk Message Queuing Adapter?

BizTalk Message Queuing (also known as MSMQT) works the same as Microsoft Message Queuing from a network perspective. The main difference is that instead of sending messages to a queue, you send them to a receive location in BizTalk Message Queuing that is a virtual representation of a Microsoft Message Queuing queue that is tied directly to the BizTalk MessageBox database. The receive locations and send ports are accessible in the BizTalk Server Administration console and BizTalk Explorer.

The value that the BizTalk Message Queuing adapter provides is the integration of its message store with the MessageBox database, a database designed to scale out on multiple servers and serve as a central repository for all user data. The BizTalk Message Queuing adapter also provides the following subset of Message Queuing functionality:

- **Binary protocol.** Support for the native (MSMQ 1.0) wire protocol.
- **Private queues.** Because messages in BizTalk Server are stored in the MessageBox database, a queue in BizTalk Message Queuing is a virtual entity. However, the support queues in its programming interface and remote servers cannot distinguish a Microsoft Message Queuing server from a BizTalk Message Queuing server. The BizTalk Message Queuing adapter supports sending and receiving messages into private queues (queues that are not published as Microsoft Active Directory directory service objects), as well as

sending messages into remote private queues, but does not support receiving messages into local public queues. This private queue support extends to send port groups as well. Send port groups cannot have public queues.

- **Workgroup mode.** The BizTalk Message Queuing adapter supports direct messaging to private queues without integration or dependency on Active Directory.

- **Active Directory mode.** BizTalk Message Queuing supports DIRECT addresses. With Active Directory it also supports PRIVATE (send/receive) and PUBLIC (send-only) addressing.

- **Transactional and best effort quality of service.** The BizTalk Message Queuing adapter supports guaranteed delivery, which is end-to-end reliability, exactly once and in order delivery.

Additional information about Microsoft Message Queuing features supported by BizTalk Message Queuing can be found in the MSMQ Frequently Asked Questions document at http://go.microsoft.com/FWLink/?LinkID=25024.

The BizTalk Message Queuing adapter provides support for transactional messaging, once-only delivery, and large messages.

You achieve support for transactional messaging, which is a stream of messages sent in order once, by synchronizing and coordinating multiple BizTalk Message Queuing adapter instances in the group. Because different BizTalk Message Queuing adapter instances may handle multiple messages that you send to the same receive location in one stream, the BizTalk Message Queuing adapter instances maintain persistent state for the delivery engine, which BizTalk Server keeps in the BizTalk Management database.

The sequence ID, sequence number, and previous sequence number properties identify messages in a stream. When you send the first message on a stream, its previous sequence number property is set to zero. For all subsequent messages, the sequence number is the successor of the sequence number property of the previously delivered sequence number property. Likewise, you set its previous sequence number property to the sequence number of the previously delivered message. As Microsoft Message Queuing or BizTalk Message Queuing attempts to deliver messages to the destinations, some messages may expire, and as a result, the sequence number and previous sequence number properties of the messages that you still need to transmit are not consecutive. For instance, consider a stream containing two messages M1 (sequence number = 1, previous sequence number = 0), and M2 (sequence number = 5, and previous sequence number = 1). Note that messages in a stream may expire without invalidating subsequent messages sent in that stream.

The destination Microsoft Message Queuing or BizTalk Message Queuing service is responsible for sending order acknowledgments to the source Microsoft Message Queuing or BizTalk Message Queuing service, indicating the sequence number of the last received message. If the source Microsoft Message Queuing or BizTalk Message Queuing service does not receive the appropriate order acknowledgment within a certain time limit, it will retransmit the message. The Microsoft Message Queuing or BizTalk Message Queuing service drops messages that are received out-of-order by the destination, because the source Microsoft Message Queuing or BizTalk Message Queuing service will resend them.

In BizTalk Message Queuing, the orchestration can provide the sequence number and previous sequence number, as properties on the BizTalk message itself.

The BizTalk Message Queuing adapter (MSMQT) and the MSMQ adapter offer different features. The following table highlights some of the differences between the two adapters.

MSMQT	Adapter for MSMQ
Delivers messages in order	Delivers messages in order
Uses only Message Queuing 2.0	Uses either Message Queuing 2.0 or 3.0
Provides large message support by streaming directly to the BizTalk MessageBox (not as memory intensive as the Adapter for MSMQ)	Provides large message support by breaking the message into parts, accumulating the parts in memory, and delivering the parts in order to the destination (more memory intensive than MSMQT)
Primarily provided for backward compatibility for existing BizTalk Server 2004 solutions built around MSMQT	Provides better performance than MSMQT
	Enables multiple applications to use the MSMQ service

In This Section

- Deprecation of the MSMQT Adapter in BizTalk Server 2006
- Migrating from the MSMQT Adapter to the MSMQ Adapter
- MSMQT Send Adapter
- MSMQT Receive Adapter
- MSMQT Protocols
- MSMQT Party Resolution
- Large Message Support in the MSMQT Adapter
- Scale Out of BizTalk Message Queuing
- Security Recommendations for the MSMQT Adapter
- BizTalk Message Queuing-MQSeries Bridge
- MSMQ Application Migration to BizTalk Server 2006
- Using the MSMQT Adapter in Active Directory Mode

Deprecation of the MSMQT Adapter in BizTalk Server 2006

The BizTalk Message Queuing adapter is deprecated in BizTalk Server 2006 and is provided for backward compatibility with existing solutions developed with BizTalk Server 2004. Because the BizTalk Message Queuing adapter is unchanged from BizTalk Server 2004 to BizTalk Server 2006, it does not use new functionality that is available to other integrated adapters. The features in BizTalk Server 2006 that do not apply to the BizTalk Message Queuing adapter are listed below.

Send Port Options in the BizTalk Server Administration Console

- The **Ordered delivery** option in the **Transport Advanced Options** pane of the **Send Port Properties** dialog box has no effect. The BizTalk Message Queuing adapter always uses ordered delivery.

- The **Stop sending subsequent messages on current message failure** option under **Ordered delivery** in the **Transport Advanced Options** pane of the **Send Port Properties** dialog box does not apply to the BizTalk Message Queuing adapter. The BizTalk Message Queuing adapter always stops sending subsequent messages on current message failure.

- The **Generate error report for failed message** option in the **Transport Advanced Options** pane of the **Send Port Properties** dialog box does not apply to the BizTalk Message Queuing adapter. Messages sent through the BizTalk Message Queuing adapter cannot be routed on failure.

Receive Port Options in the BizTalk Server Administration Console

The **Generate error report for failed message** option in the **General** pane of the **Receive Port Properties** dialog box does not apply to the BizTalk Message Queuing adapter. Messages received by the BizTalk Message Queuing adapter cannot be routed on failure.

Receive Location Options in the BizTalk Server Administration Console

The **RecoverableInterchangeProcessing** option in the **Configure Pipeline** dialog box does not apply to the BizTalk Message Queuing adapter. Messages received by the BizTalk Message Queuing adapter cannot use recoverable interchange disassembly.

Migrating from the MSMQT Adapter to the MSMQ Adapter

This topic discusses points to consider regarding end-to-end ordered delivery, transactional consistency, high availability, and scalability before migrating solutions from the BizTalk Message Queuing (MSMQT) adapter to the Message Queuing (MSMQ) adapter. For purposes of this topic ordered delivery, transactional consistency, high availability, and scalability are defined as follows:

- **Ordered delivery.** Guarantee that messages are sent out of BizTalk Server in the same order that they are received.

- **Transactional consistency.** Guarantee that messages being processed are not lost or duplicated due to hardware, software, or network failure.

- **High availability.** Guarantee that services used to process messages are always available for processing.

- **Scalability.** Ability to increase existing message-processing capacity.

Migration of Solutions Built Around MSMQT Is Not Required

There is no requirement to migrate existing solutions from the MSMQT adapter to the MSMQ adapter. The MSMQT adapter is present and fully supported in BizTalk Server 2006. The MSMQT adapter is deprecated, but that only means that it may not be available in subsequent versions of BizTalk Server.

End-to-End Ordered Delivery

The MSMQT adapter ensures end-to-end ordered delivery of messages. This means that if an MSMQ application sends messages 1, 2, and 3 to a receive location bound to the MSMQT adapter, then these messages are delivered to an orchestration or send port in BizTalk Server in the same order: 1, 2, 3. One use of this functionality might include stock market trades that must be submitted and executed in the same order that they are received.

End-to-end in-order delivery with MSMQT requires many components to be working together. The following sequence of events illustrates how this is done with the MSMQT adapter:

1. The MSMQ API on a remote computer receives messages 1, 2, and 3 in order and pushes them into a transactional, local queue in the same order: 1, 2, 3.

2. An MSMQ client on the remote computer takes the messages from the queue and sends them to the MSMQT queue in order: 1, 2, 3.

3. The MSMQT adapter receives the messages in order 1, 2, 3 and hands them to the BizTalk MessageAgent component in the same order: 1, 2, 3.

4. The BizTalk MessageAgent component ensures that the messages are sent to the MessageBox in order: 1, 2, 3.

5. The MessageBox routes the messages and ensures that if they are routed to the same instance of an orchestration or a send port, that they are delivered to this instance in the same order: 1, 2, 3.

In BizTalk Server 2004, MSMQT is the only adapter capable of guaranteeing end-to end ordered delivery. All of the other integrated BizTalk adapters can potentially change the order of the messages in steps 3 through 5 listed above. Most of the other integrated adapters complete step 3 by the use of a component known as the End Point Manager, which is inherently multithreaded and therefore does not preserve order. The MSMQ adapter for BizTalk Server 2004 can use a "Serial Processing" feature that preserves order for step 3, but

it does not then ask the MessageAgent to preserve order going forward, so the messages may be routed to an orchestration or send port out of order.

End-to-end ordered delivery in BizTalk Server 2006 with the MSMQ adapter

To achieve end-to-end ordered delivery with the MSMQ adapter in BizTalk Server 2006, follow these steps:

1. Enable the **Ordered Delivery** property on the receive port for the subscribing orchestration or send port. In BizTalk Server 2006 receive ports in orchestrations and send ports have an **Ordered Delivery** configuration option. If this option is enabled, then the orchestration receive port or the send port asks the MessageBox to deliver messages to it in the same order they were submitted into the MessageBox.

2. Set the **Ordered Processing** property for the receive location that is bound to the MSMQ adapter to **True**. In BizTalk Server 2006, receive locations that use the MSMQ transport can be configured to use Ordered Processing which, if enabled, ensures that messages are sent to the MessageBox in the same order that they were received.

3. Set the **Transactional** property for the receive location that is bound to the MSMQ adapter to **True**.

4. Ensure that any MSMQ queues that are being monitored by the MSMQ receive locations are marked as "Transactional". This property must be set on the queues to ensure ordered delivery of messages.

Transaction Usage When Processing Messages with the MSMQT Adapter vs. the MSMQ Adapter

With regard to transaction usage, there is a key difference between how the MSMQT and MSMQ adapters process messages.

When using the MSMQT adapter, the process of receiving a message from the network and processing it with BizTalk Server is handled under a single transaction. When using the MSMQT adapter, ACK messages generated for the sender are an indication that the message has been received and has been successfully processed by BizTalk Server.

When using the MSMQ adapter, the processes of receiving a message from the network and processing it with BizTalk Server are handled under two separate transactions, one for receiving from the network and one for processing it with BizTalk Server. When using the MSMQ adapter, ACK messages generated for the sender are only an indication that the message has been successfully received from the network, not that the message has been successfully processed by BizTalk Server. The sender receives an ACK from the Microsoft Message Queuing server when the message is received from the network and persisted into the local MSMQ queue regardless of whether BizTalk Server is installed. After the message has been persisted to the MSMQ queue, the BizTalk MSMQ adapter picks it up, processes it, and publishes it to the MessageBox. If this process fails, the message is either sent to the BizTalk suspended queue or left in the local MSMQ queue (when using transactional processing), and the sender has no indication that the message failed processing in BizTalk Server.

If your architecture requires that you receive ACKs when your messages are successfully processed by BizTalk Server, then you must add application-level ACKs if you are migrating from the MSMQT adapter to the MSMQ adapter. You will need to update your orchestration and sending application to implement application-level ACKs.

High Availability (Transactional, in Order)

To provide high availability for the MSMQT adapter you can either add multiple computers to the receive host and configure Network Load Balancing (NLB) for fault tolerance or, in BizTalk Server 2006, you can cluster the default BizTalk Host. If you are running the MSMQT adapter in conjunction with NLB, if one server goes down the other servers handle the load. If you are running the MSMQT adapter handlers on a clustered host, if one host node fails, cluster software fails over the clustered host to the other node. When using the MSMQ adapter, NLB does not work if you need transactional processing with no data loss because the MSMQ adapter uses local MSMQ queues for intermediate storage. In this scenario, if a message has been delivered to the local MSMQ queue but has not been consumed by the MSMQ adapter, the message is lost if the computer fails.

To provide high availability and transactional consistency with the MSMQ adapter on BizTalk Server 2006 you should do the following:

1. Configure Microsoft Message Queuing (MSMQ) as a clustered resource in a Windows Server cluster group on your BizTalk servers.

2. Configure the MSMQ adapter receive handler in a BizTalk Host instance that has been configured as a cluster resource in the same cluster group as the clustered MSMQ resource.

3. Configure the cluster resource for the BizTalk Host instance so that it maintains a dependency on the clustered MSMQ resource.

To implement ordered delivery with this architecture, follow the steps presented earlier under "End-to-end ordered delivery in BizTalk Server 2006 with the MSMQ adapter."

High Availability (Nontransactional, Not in Order)

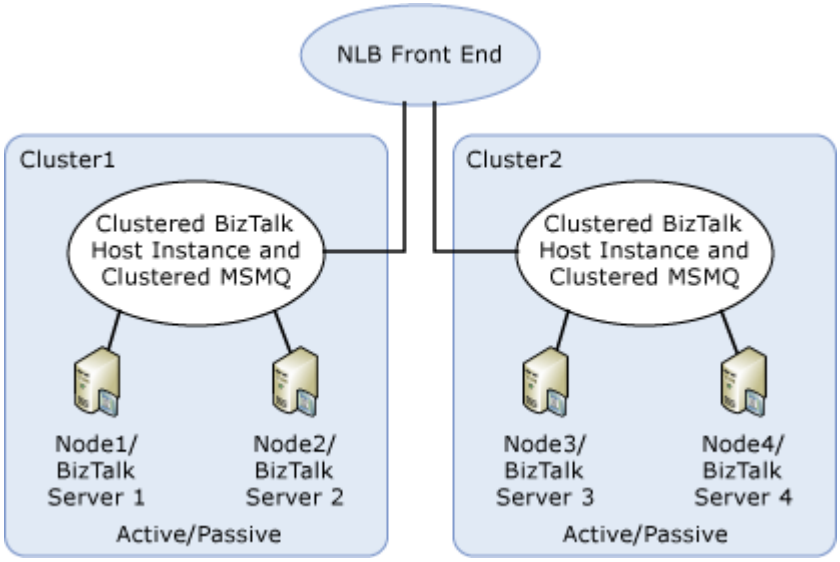
If you need high availability but do not need transactional processing, you can achieve this with the MSMQ adapter by implementing NLB and running instances of a host configured with the MSMQ send and receive handlers on multiple BizTalk servers behind NLB. When implementing NLB with MSMQ you should follow best practices as documented in Microsoft Knowledge Base article 899611, "How Message Queuing can function over Network Load Balancing (NLB)" available at http://go.microsoft.com/fwlink/?LinkId=57510. In this scenario, if one of the BizTalk servers fails, the messages that are running in the host instance on that BizTalk server will be unavailable until the BizTalk server is recovered. This configuration provides high availability because if one of the BizTalk servers is not available, NLB routes requests to the other BizTalk server.

Scalability (Nontransactional, Not in Order)

You can achieve scalability by following the guidelines for high availability (nontransactional) and adding additional host instances. This architecture provides fast delivery as well as scalability, but does not provide ordered delivery.

Scalability (Transactional, Not in Order)

For transactional delivery of messages without ordered delivery, you can combine the use of NLB with MSMQ and Windows clustering. This architecture requires that you configure at least two cluster groups on two separate Windows Cluster environments following the steps under "High Availability (Transactional, in Order)" for each Windows Cluster. You then implement NLB to distribute load between the cluster groups. Because Windows NLB is not supported running on a Windows Cluster, this scenario requires a hardware NLB solution. The following diagram illustrates this architecture.



Scalability (Transactional, in Order)

Because the MSMQ API does not support remote transactional reads, scaling out an architecture that provides high availability, transactional consistency, and ordered delivery is problematic. To scale out, multiple local MSMQ queues must be used. If the TCP/IP connection for the MSMQ session to the NLB is broken, it may subsequently be restored by NLB to a different computer, which can cause out-of-order delivery.

One possible workaround to this limitation is to manually load balance message delivery by allocating destination queues to different computers. You can so this by tying specific BizTalk Host instances to specific MSMQ queues. So for example, if you receive a large number of documents from one particular trading partner, create a separate host and receive queue on a particular BizTalk server for just that trading partner.

Summary

The following table summarizes the architectures that you can implement to accommodate specific functionality.

Functionality	Neither NLB nor cluster	NLB	Cluster	NLB and cluster
End-to-end ordered delivery	Yes	No	Yes	Possible with manual configuration
Transactional consistency	No (messages can be lost or duplicated if service failure occurs)	No	Yes	Yes
Highly available	No	Yes	Yes	Yes
Scalable	No	Yes	No	Yes

MSMQT Send Adapter

The BizTalk Message Queuing send adapter natively processes only DIRECT format names.

Messages sent by the BizTalk Message Queuing adapter contain byte order marks. Therefore, to read messages sent from MSMQT to traditional Message Queuing from .NET Framework applications you have to access the message body stream.

The following code example illustrates how to access the message body stream:

The following code shows a simple string:

Alternatively, you can remove the byte order mark from documents processed by the BizTalk Message Queuing adapter with a custom send pipeline. To remove the byte order mark, create a send pipeline that uses the **XML Assembler** or **Flat File Assembler** component with the **Preserve byte order mark** property set to **False** and specify this pipeline in the Send port that is bound to the MSMQT adapter. For more information about creating pipelines, see Creating Pipelines with Pipeline Designer .

The BizTalk Message Queuing adapter does not process non-direct format names, but you can use a native message queuing router for this purpose. If you have configured the name of the message queuing router, and if the router is available, the BizTalk Message Queuing adapter passes the non-direct outgoing message to the router, which performs all of the processing. You can set the router name in the property page of the BizTalk Message Queuing adapter. For more information, see How to Configure an MSMQT Send Handler . For more information about format names, see http://go.microsoft.com/FWLink/?LinkID=24504.

MSMQT Receive Adapter

You use the BizTalk Message Queuing receive adapter to receive messages into a BizTalk message queue that is a virtual representation of a Microsoft Message Queuing queue. The BizTalk message queue is tied directly to the BizTalk MessageBox database and therefore is

not subject to Microsoft Message Queuing limitations on the size or number of messages that can be stored.

MSMQT Protocols

The BizTalk Message Queuing adapter supports the Microsoft Message Queuing binary (native) protocol. It does not support the following features:

- Multicast (provided by Message Queuing)
- Remote read
- SRMP (Microsoft Message Queuing over HTTP)
- Microsoft Message Queuing protocol-level encryption

The BizTalk Message Queuing adapter can send messages using PUBLIC and PRIVATE formats only with the help of a Microsoft Message Queuing router. However, you can still access public and private queues using the DIRECT format. You can send messages to BizTalk Message Queuing using only DIRECT and PRIVATE formats. PUBLIC is not supported.

MSMQT Party Resolution

The BizTalk Message Queuing adapter communicates with the Party Resolution pipeline component. The adapter provides both the certificate, which the adapter uses by default to resolve the party, and the security identifier (SID) to the pipeline component. However, when you install BizTalk Message Queuing in Active Directory mode, it may be easier to resolve the party based on the user name. To do this, create a custom pipeline with a Party Resolution pipeline component, and then configure the Party Resolution component to use the SID when resolving the party. Set the **Resolve Party By Certificate** property to **False**, and the **Resolve Party By SID** property to **True**. For more information, see How to Configure the Party Resolution Pipeline Component .

Large Message Support in the MSMQT Adapter

Native message queuing cannot process a message with a body larger than 4 megabytes (MB). However, BizTalk Server includes an add-on for native message queuing that permits processing messages larger than 4 MB. BizTalk Server delivers this add-on as the MQRTLarge.dll file in the BizTalk Server SDK, and exposes the **MQSendLargeMessage** and **MQReceiveLargeMessage** APIs and the analogous COM model. These functions are implemented as standard message queuing APIs, **MQSendMessage** and **MQReceiveMessage**, respectively.

To participate in large message exchanges, the message queuing computer must have the MQRTLarge.dll file installed, and the message queuing application must use the add-on APIs. Otherwise, complete messages are fragmented.

A few notes about large messages:

- Processing large messages prevents use of the **BizTalk Message Queuing.Extension** property, configurable through XLANG/s.

- When sending large messages, make sure that the %temp% drive has enough free space to store all of the messages processed by the BizTalk Message Queuing adapter. Large messages are temporarily stored on the disk until the destination acknowledges them.

For more information about the MQRTLarge.dll file, see Large Message to MSMQT .

Scale Out of BizTalk Message Queuing

To increase availability of BizTalk messaging queues, you can deploy BizTalk Server on a group of servers. To scale out, you add nodes (physical computers) to the group of servers. The server group can host multiple instances of BizTalk services that share the same back-end message store, the MessageBox database.

You can implement such a server group on a Network Load Balancing (NLB) system where multiple server nodes share one virtual name and IP address.

When configuring the BizTalk Message Queuing transport on an NLB system with the BizTalk Configuration tool, use the following guidelines:

- If Microsoft Message Queuing (MSMQ) is not installed and you do not plan to install Message Queuing on any of the server nodes, then leave the default option of **Bind Message Queuing to all IP addresses on this computer** selected when configuring the BizTalk Message Queuing transport. If the BizTalk server has a single static IP address you can optionally select the option to **Bind to this IP address only**.

- Specify the NLB virtual server name for the **Computer name** entry on each BizTalk server that is participating in the NLB cluster.

The use of NLB improves throughput under the following conditions:

- Multiple clients are accessing multiple BizTalk message queues

- A single client is accessing multiple BizTalk message queues.

Security Recommendations for the MSMQT Adapter

The following is a list of security recommendations for the BizTalk Message Queuing adapter:

- The BizTalk Message Queuing adapter runtime requires at least User (Domain User or Local User) privileges. It is recommended that you do not run with Administrator privileges.

- The BizTalk Message Queuing adapter setup requires Administrator privileges.

- The BizTalk Message Queuing adapter does not contain any custom components, but it does call the BizTalk Messaging Engine, which in turn calls pipelines. The pipelines may be custom pipelines, in which case you should apply the security recommendations for the pipeline to the BizTalk Message Queuing adapter.

- It is recommended that you use the BizTalk Message Queuing adapter in authentication-required mode. Set the **Requires MSMQ authentication** flag on the receive location and **AuthenticationRequired** (Drop messages) on the receive port to true. These settings help prevent Denial of Service attacks. Keep in mind that the **AuthenticationRequired** flag on the receive port requires the Party Resolution pipeline component to be configured correctly, and that the parties are defined in BizTalk Explorer. For more information about configuring the Party Resolution pipeline component, see MSMQT Party Resolution .

- Generic security recommendations for Microsoft Message Queuing are applicable to BizTalk Message Queuing as well. For more information, see Microsoft Message Queuing Help at http://go.microsoft.com/FWLink/?LinkID=26104.

BizTalk Message Queuing-MQSeries Bridge

The BizTalk Message Queuing-MQSeries bridge works the same as the MSMQ-MQSeries bridge with few exceptions. For information about how the MSMQ-MQSeries bridge works, see the Microsoft Host Integration Server (HIS) 2000 Help at http://go.microsoft.com/fwlink/?linkid=16280.

The following are important reminders for MSQMT-MQSeries bridge development:

- You must have HIS Service Pack 1 installed on the computer.

- Install BizTalk Message Queuing in Active Directory-integrated mode, because you configure bridge interaction through Active Directory.

- Configure the MQSeries bridge as explained in the Host Integration Server documentation. Test your configuration using the MQSeries send and receive tools provided with Host Integration Server. For more information, see "MSMQ-MQSeries Bridge Setup and Configuration" in the Host Integration Server documentation.

- To send a message to an MQSeries queue, set the BizTalk Message Queuing port address to PUBLIC=<MQSeries Queue GUID>.

  - To obtain the MQSeries queue GUID, go to Active Directory Sites and Services and open the public queues of the computer that represents your MQSeries server. The GUID appears on the queue property pages.

  - You cannot use a DIRECT= format name to send messages to MQSeries.

- The name of the BizTalk Message Queuing and Microsoft Message Queuing queues must be less than 25 characters and in all uppercase characters.

- On the MQSeries side, you must use the following format for BizTalk Message Queuing queues: DIRECT_OS/<computer>/P_/queue.

MSMQ Application Migration to BizTalk Server 2006

This topic describes how to migrate Microsoft Message Queuing applications from BizTalk Server 2006 to BizTalk Server 2006.

Applications Exchanging Messages Smaller Than 4 MB

If your application uses Microsoft Message Queuing protocol-level encryption, you need to modify it to implement the encryption at the application level. If you are implementing application-level encryption, consider implementing one of the encryption methods supported by the BizTalk MIME/SMIME pipeline component. For more information about the MIME/SMIME pipeline component, see MIME/SMIME Decoder Pipeline Component or MIME/SMIME Encoder Pipeline Component .

BizTalk Server does not support multicast and Microsoft Message Queuing messages over an HTTP transport. However, you can still use them if you write another Microsoft Message Queuing application that reads the multicast and HTTP messages and then sends them to BizTalk Server using the DIRECT format.

Microsoft Message Queuing to BizTalk Message Queuing Considerations

- In some cases, to send messages from Microsoft Message Queuing to BizTalk Message Queuing, you must modify your code. If you are using a PUBLIC format name in your application, you must replace it with a DIRECT or PRIVATE format name.

- If your application is using the path name to send messages, you must change the code to use the format name and one of the DIRECT or PRIVATE format names.

- If you use the PRIVATE format name, you must have a Microsoft Message Queuing router in your domain and configure BizTalk Message Queuing to use it.

BizTalk Message Queuing to Microsoft Message Queuing Considerations

- BizTalk Server supports all formats, but you need to configure a router for PUBLIC and PRIVATE.

Applications Exchanging Messages Larger Than 4 MB

Microsoft Message Queuing does not natively support large messages. To use large messages in your Microsoft Message Queuing application, it is likely that you will have to rewrite the code. The BizTalk Server 2006 Software Development Kit (SDK) contains the file MQRTLarge.dll, which enables BizTalk Message Queuing to support large messages. For more information about the MQRTLarge.dll file, see Large Message Support in the MSMQT Adapter . There are no COM or managed interfaces for large messages, so your code must be in C or C++.

For more information, see the restrictions listed earlier in "Applications Exchanging Messages Smaller Than 4 MB."

Writing to the Message Body Stream

By default, when you pass your messages through the new BizTalk Message Queuing adapter, the .NET Framework wraps the message body in XML tags. Because of these extra tags, the message will no longer match your schema. To overcome this problem you need to write directly into the message body stream, as shown in the following code:

Using the MSMQT Adapter in Active Directory Mode

The following tips apply to the BizTalk Message Queuing adapter in Active Directory mode:

- Configuring Active Directory integration and setting the computer name for BizTalk Message Queuing is only possible during installation. Running the configuration wizard again after the BizTalk service has started overwrites the existing configuration. If you need to change these parameters, reinstallation is the only option

    - If you want to set up BizTalk Message Queuing in Active Directory mode, you need permissions to create, delete, and modify Active Directory computer objects. Specifically, you need permissions to add and delete all children on computers, and the Active Directory computer. You also need permissions to add and delete all children on sites and servers on Active Directory Sites and Services to enable the routing option for Microsoft Message Queuing. Typically, only domain administrators have these privileges.

- Local accounts do not have rights to access Active Directory, Therefore, sending to public and private queues or using authentication does not work when the BizTalk service is running under a local account.

How to Install Microsoft Message Queuing and BizTalk Message Queuing Side-by-Side

BizTalk Message Queuing (MSMQT) and Microsoft Message Queuing (also known as MSMQ) can be installed side-by-side. Use the following guidelines to ensure a successful side-by-side installation.

To install Microsoft Message Queuing and BizTalk Message Queuing side-by-side

1. Ensure that your operating system supports side-by-side installation. Only Microsoft Windows Server 2003, Windows 2000 Server Service Pack 4, and Windows XP Service Pack 2 support side-by-side installation.

2. You will need two IP addresses on every computer that will run BizTalk Message Queuing and Microsoft Message Queuing side-by-side. You can install two network adapters on these computers or add two static IPs on the same network interface card. If you choose the second option, both addresses must be in the same subnet. Microsoft

Message Queuing will use the first IP and BizTalk Message Queuing will use the second IP.

3. In addition to having two IP addresses you must have two computer names for every computer that will run BizTalk Message Queuing and Microsoft Message Queuing side-by-side. For example, your BizTalk Server might already be named "BizTalk" and you could append "MSMQT" to your computer name to add the second computer name "BizTalkMSMQT"

4. For demonstration purposes, assume that your computer name is MyServer and BizTalk Server Message Queueing will use MyServerMSMQT.

5. Ask your network administrator to configure the DNS server so that a query for MyServer will always return IP1 and a query for MyServerMSMQT will always return IP2. You can achieve this by disabling DNS registration on your computers and manually adding entries in the DNS tables.

6. Test the DNS configuration. At the command prompt, type **nslookup**, and press ENTER.

7. At the command prompt, type **ping MyServer**, and verify that the system only returns IP1.

8. At the command prompt, type **ping MyServerMSMQT**, and verify that the system only returns IP2.

9. Install Microsoft Message Queuing.

10. Configure Microsoft Message Queuing for side-by-side use. Using the Registry Editor, browse to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters**. Add a new string value called **BindInterfaceIP**, set its value to **IP1**, and then restart the Microsoft Message Queuing service.

11. Install BizTalk Server. In the **Configuration Framework MSMQT** screen, set the **IP** field to **IP2**. Set the **Computer name** to **MyServerMSMQT**. If you plan to use PUBLIC or PRIVATE format names, select the **Active Directory** box. Do not select the **Register in dns** box unless advised to do so by the network administrator.

Configuring the MSMQT Adapter

This section describes how to configure the BizTalk Message Queuing adapter.

In This Section

- How to Modify the Default Configuration of the MSMQT Adapter
- How to Register and Unregister the MSMQT Adapter
- How to Configure an MSMQT Receive Handler
- How to Configure an MSMQT Receive Location

- How to Configure an MSMQT Send Handler
- How to Configure an MSMQT Send Port
- MSMQT Configuration and Tuning Parameters
- MSMQT Adapter Property Schema and Properties

How to Modify the Default Configuration of the MSMQT Adapter

The BizTalk Message Queuing adapter is installed on BizTalk Server 2006 with a default configuration. The BizTalk Message Queuing adapter default configuration is applied even if the adapter is not specifically configured with the BizTalk Configuration program. Therefore, the BizTalk Message Queuing adapter will provide full functionality if it is added to the list of adapters in the BizTalk Administration console, even if it has not been configured with the BizTalk Configuration program.

To modify the default configuration of the BizTalk Message Queuing adapter

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Configuration** to launch the BizTalk Configuration program.
2. In the right pane of the **Microsoft BizTalk Server Configuration** dialog box double-click the MSMQT feature to display the **BizTalk Message Queuing Transport** configuration page.
3. Apply the appropriate options to modify the default configuration. See **Configuration Wizard, MSMQT Page** for a description of each option.

How to Register and Unregister the MSMQT Adapter

The BizTalk Message Queuing adapter is included in the BizTalk Server 2006 source code, but for security reasons, BizTalk Server does not register it by default during the product configuration. To use the BizTalk Message Queuing adapter, you must first register it using the BizTalk Server Administration console. After registration, the adapter appears in drop-down menus along with the other native adapters.

Under certain circumstances you may want to unregister the BizTalk Message Queuing adapter. The procedure for unregistering the BizTalk Message Queuing adapter is described below.

For more information about security privileges required to add an adapter in the BizTalk Server Administration console, see **Minimum Security User Rights**.

To register the BizTalk Message Queuing adapter

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, and expand **Platform Settings**.

3. Right-click **Adapters**, click **New**, and then click **Adapter**.

4. In the **Add Adapter** dialog box, do the following.

Use this	To do this
Name	Type **BizTalk Message Queuing**.
Adapter	Select **MSMQT** from the drop-down list.

5. Click **OK**.

   The static adapter now appears in the list of adapters in the right window of the BizTalk Server Administration console. Before using the adapter, you must stop and restart the host instance.

To stop and restart the host instance

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then select **Host Instances**.

2. In the right pane, right-click the host instance (typically, **BizTalkServerApplication**), and then click **Stop**.

   The status of the host instance changes to **Stopped**.

3. In the results pane, right-click the host instance, and then click **Start**.

   The status of the host instance changes to **Running**.

To unregister the BizTalk Message Queuing adapter

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

   The list of adapters appears under the folder.

3. Right-click the **BizTalk Message Queuing** adapter and then click **Delete**. Click **Yes** in the **Confirm adapter delete** dialog box.

How to Configure an MSMQT Receive Handler

Use the following procedure to change the host with which the BizTalk Message Queuing receive handler is associated.

To configure the host for a BizTalk Message Queuing receive handler

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

   The list of adapters appears under the folder.

2. Click **BizTalk Message Queuing,** in the right pane right-click the receive handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.

4. Click **OK**.

How to Configure an MSMQT Receive Location

You can set BizTalk Message Queuing receive location adapter variables in the BizTalk Server Administration console. If properties are not set in the receive location, the default receive handler values set in the BizTalk Server Administration console are used.

To configure variables for a BizTalk Message Queuing receive location

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application in which you want to create a receive location.

2. In the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties** to display the **Receive Port Properties** dialog box.

3. Select the **Receive Locations** option from the left pane of the **Receive Port Properties** dialog box, and in the right pane double-click an existing receive location or click **New** to create a new receive location. This displays the **Receive Location Properties** dialog box.

4. Under the **Transport** section next to **Type**, select **MSMQT** from the drop-down list, and then click **Configure** to configure the transport properties for the receive location.

5. In the **BizTalk Message Queuing Transport Properties** dialog box, do the following:

Use this	To do this
Queue name	Identify the name of the queue. BizTalk Message Queuing queue names need to be less than 25 characters if you want to receive MQSeries messages. Use all uppercase letters for the BizTalk Message Queuing queue names that receive MQSeries messages. If there is no queue defined for an incoming message, BizTalk Server rejects the message as **No Such Queue**.
Requires MSMQ Authentication	Indicate whether Message Queuing (also known as MSMQ) authentication is required. If selected, the adapter rejects all messages that do not use Message Queuing protocol-level authentication.
Non-Transactional Queue	Indicate whether the queue is transactional. If selected, the queue only accepts nontransactional messages. Select this box to receive acknowledgment messages in this queue.
Queue number	Specify the address for the queue using the Message Queuing private format name. This number is populated by default. When installed in Active Directory mode, BizTalk Message Queuing can receive messages that were sent using the PRIVATE=<BizTalk Message Queuing computer GUID>\<queue number> format name. The queue number must be greater than or equal to four. Numbers one through three are reserved for standard use.

6. Click **OK**.

7. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and then click **OK**.

How to Configure an MSMQT Send Handler

You can globally configure the BizTalk Message Queuing send handler by using the BizTalk Server Administration console.

To configure global variables for a BizTalk Message Queuing send handler

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded list of adapters, select the configured BizTalk Message Queuing adapter. The list of send and receive handlers that are bound to the configured BizTalk Message Queuing adapter appears in the right pane.

3. In the right pane, double-click the send handler to open the **Adapter Handler Properties** dialog box.

4. In the **Adapter Handler Properties** dialog box, in the **Host Name** list, select the host with which the send handler will be associated.

5. Click **Properties** to display the **MSMQT Transport Properties** dialog box.

6. Enter a value for **MSMQ Router name** if you will be sending messages using non-DIRECT format names.

7. Click **OK** and click **OK** again to close the **Adapter Handler Properties** dialog box.

How to Configure an MSMQT Send Port

You can set BizTalk Message Queuing send port adapter variables either programmatically or by using the BizTalk Server Administration console. If properties are not set for the send port, the default send handler values set in the BizTalk Server Administration console are used.

How to Configure an MSMQT Send Port Programmatically

For configuring the BizTalk Message Queuing adapter programmatically, the following table lists the properties used by the adapter in the <xs:element name="InboundTransportType" type="xs:string" /> syntax.

You must configure the adapter in the default host.

Namespace	Property	Description
System	**ExpirationTime**	DateTime.
System	**InboundTransportLocation**	String Destination queue for incoming message.
System	**InboundTransportType**	String Incoming message will contain "MSMQ" if the message came through the BizTalk Message Queuing adapter.
System	**OutboundTransportLocation**	String Destination queue for outgoing message.

BizTalk Message Queuing	**Ack**	String
BizTalk Message Queuing	**AdminQueue**	String
BizTalk Message Queuing	**AppSpecific**	Integer
BizTalk Message Queuing	**ArrivedTime**	DateTime
BizTalk Message Queuing	**AuthLevel**	Boolean If set to true, authentication will be used.
BizTalk Message Queuing	**Class**	Short
BizTalk Message Queuing	**Extension**	HexBinary
BizTalk Message Queuing	**CorrelationID**	String in the GUID\number format (For example, "{48A2DE3E-ECB4-4F5D-9A48-50A362F5A418}\\28"). Correlation ID for the incoming message.
BizTalk Message Queuing	**HashAlgorithm**	Integer. The **System.Messaging.HashAlgorithm** enumeration defines the valid values.
BizTalk Message Queuing	**InboundResponseQueue**	String Response queue for incoming message.
BizTalk Message Queuing	**IsAuthenticated**	Boolean Shows whether the incoming message was signed.
BizTalk Message	**IsFirstInTransaction**	Boolean

Queuing		Determines where the transaction starts.
BizTalk Message Queuing	**IsLastInTransaction**	Boolean Determines where the transaction ends.
BizTalk Message Queuing	**IsXactMessage**	Boolean
BizTalk Message Queuing	**Label**	String
BizTalk Message Queuing	**MsgID**	Integer Number part of the incoming MSGID of the message.
BizTalk Message Queuing	**ResponseQueue**	String Response queue for the outgoing message.
BizTalk Message Queuing	**Priority**	Integer
BizTalk Message Queuing	**PrivLevel**	Integer
BizTalk Message Queuing	**SentTime**	DateTime
BizTalk Message Queuing	**SourceMachineGUID**	String GUID part of the incoming MSGID of the message.
BizTalk Message Queuing	**TimeToReachQueue**	Integer Time to reach the queue (in milliseconds).
BizTalk Message Queuing	**TransactionID**	String

How to Configure an MSMQT Send Port with the BizTalk Server Administration Console

To configure the adapter by using the BizTalk Server Administration console, use the following procedure.

To configure variables for a BizTalk Message Queuing send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. See How to Create a Send Port for more information. Configure all of the send port options and specify **MSMQT** for the **Type** option in the **Transport** section of the **General** page.

2. On the **General** page, in the **Transport** section, click the **Configure** button next to **Type** to display the **MSMQT Transport Properties** dialog box.

3. In the **BizTalk Message Queuing Transport Properties** dialog box, do the following:

Use this	To do this
Destination Queue	Indicate the Message Queuing (also known as MSMQ) or BizTalk Message Queuing destination queue. The value may not be a public address. Examples of valid values include: **DIRECT=TCP:172.12.22.11\Private$\<QueueName>** and **DIRECT=OS:<machineName>\Private$\<QueueName>**
Use MSMQ Authentication	Identify whether BizTalk Message Queuing uses protocol authentication every time it sends a message on this port.

4. Click **OK** and **OK** again to save settings.

MSMQT Configuration and Tuning Parameters

Most BizTalk Message Queuing configuration and tuning parameters have equivalent Microsoft Message Queuing parameters. Some parameters, such as **MsmqtMachineName**, are specific to BizTalk Server and enable side-by-side Microsoft Message Queuing and BizTalk Message Queuing installations. The following table contains the complete list of BizTalk Message Queuing parameters and their locations in the registry.

Key name	Type	Default	Unit	Explanation
MsmqtBoxSize	ULONG	1000	msgs	Defines the size of the message container used message references in the instance state. Read-only. BizTalk Message Queuing only.
MsmqtResendProbeSize	ULONG	1000	msgs	Defines the number of messages (from the messages-in-transit) that BizTalk Message Queu resend at each retry. BizTalk Message Queuing only.
MsmqtOutInactivityToClose	ULONG	5*60	sec	Defines the inactivity time period after which stream instance will be dehydrated.
MsmqtOutInactivityToDestroy	ULONG	24*60*60	sec	Defines the inactivity time period after which stream instance will be destroyed.
MsmqtInInactivityToClose	ULONG	5*60	sec	Defines the inactivity time period after which stream instance will be dehydrated.
MsmqtInInactivityToDestroy	ULONG	90*24*60*60	sec	Defines the inactivity time period after which B destroys the incoming stream instance.
MsmqtRetryInterval0	ULONG	30	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the first res outgoing instance.
MsmqtRetryInterval1	ULONG	30	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the second re outgoing instance.
MsmqtRetryInterval2	ULONG	30	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the third res outgoing instance.
MsmqtRetryInterval3	ULONG	5*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the fourth re outgoing instance.
MsmqtRetryInterval4	ULONG	5*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the fifth res outgoing instance.
MsmqtRetryInterval5	ULONG	5*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the sixth res

				outgoing instance.
MsmqtRetryInterval6	ULONG	30*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the seventh re outgoing instance.
MsmqtRetryInterval7	ULONG	30*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the eighth re outgoing instance.
MsmqtRetryInterval8	ULONG	30*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the ninth res outgoing instance.
MsmqtRetryInterval9	ULONG	60*60	sec	Defines the time the BizTalk Message Queuing ada an order acknowledgment before the tenth res outgoing instance, and every following time.
MsmqtFragmentSize	ULONG	1 Mb	bytes	Defines the size of the message fragments tha Message Queuing adapter divides large messages in Read-only. BizTalk Message Queuing only.
MsmqtMaxSizeToSendAsSingleMsg	ULONG	4 Mb	bytes	Defines the maximum size of a non-fragmented me Read-only. BizTalk Message Queuing only.
MsmqtOrderAckDelay	ULONG	5	sec	Defines the time to delay sending order ack message.
MsmqtBinaryStoreAckTimeout	ULONG	INF	sec	Defines the time-out period for the stored ack message of the binary protocol.
MsmqtBinarySessionAckTimeout	ULONG	INF	sec	Defines the time-out period for the session ack message of the binary protocol.
MsmqtBinarySessionCleanupTimeout	ULONG	5000*60	sec	Defines the period of inactivity after which the Biz Queuing adapter closes the binary protocol session.
MsmqtLongLivedDefaultTimeout	ULONG	4	days	Defines the default value for the **Time To L** property.
MsmqtPort	USHORT	1801	n/a	Defines the TCP/IP port that the BizTalk Mess adapter uses to listen for new messages.

MsmqtPingPort	USHORT	3527	n/a	Defines the TCP/IP port that the BizTalk Mess adapter uses for ping protocol.
MsmqtBinaryWindowSize	USHORT	64 (debug 32)	days	Defines the size of the window for message tran session for a binary protocol. BizTalk Message Queuing only.
MsmqtRetrialsToBeDeclaredPoisonMessage	ULONG	5	n/a	Defines the number of times the processing is tried Server processing for a message fails. After specified number of times, the message is internall harmful. BizTalk Message Queuing only.
MsmqtMaxBatchTimeout	WORD	2000	sec	Defines the maximum time for collecting the batch messages from the BizTalk Message Queuing lay Message Queuing. BizTalk Message Queuing only.
MsmqtMaxBatchSize	WORD	20	msgs	Defines the maximum size of the batch for deliver from the BizTalk Message Queuing layer to Biz Queuing. BizTalk Message Queuing only.
MsmqtBlockSendingReceiveFinalAck	WORD	0	0/1	Defines whether BizTalk Message Queuing se acknowledgment to the sender. Setting this proper BizTalk Message Queuing from sending acknowledgment.
MsmqtWorkgroupMode	WORD	1	0/1	Defines whether BizTalk Message Queuing is i workgroup mode. Set this property to 1 if you installed BizTalk Messa a workgroup mode (as opposed to Active Directory Read-only.
MsmqtIptPerCPU	USHORT	2	threads	Defines the number of BizTalk Message Queuing w per processor for processing incoming messages. BizTalk Message Queuing only.
MsmqtIptAdditional	USHORT	1	threads	Defines the number of additional (per-CPU) Biz Queuing worker threads for processing incoming me

				BizTalk Message Queuing only.
MsmqtIptAccumulationSize	USHORT	5000	msgs	Defines the maximum number of incomin accumulated for one worker thread. BizTalk Message Queuing only.
MsmqtRouter	LPWSTR	NULL	n/a	Defines the name of the Microsoft Message Qu computer to use for sending messages with a non- name.
MsmqtBindingIP	LPWSTR	INADDR_ANY	n/a	Defines the IP address to use, which may be diffe default computer IP address for Active Directory i load balancing. Read-only.
MsmqtMachineName	LPWSTR	<local computer>	n/a	Defines the computer name for Active Directory and Read-only. BizTalk Message Queuing only.
MsmqtDomain	LPWSTR	<computer domain>	n/a	Defines the domain name where you have re message queuing computer. Read-only. BizTalk Message Queuing only.

Some of parameters cannot be set after the system starts the BizTalk Message Queuing service. These parameters are marked as read-only. Parameters that do not have equivalent parameters in Microsoft Message Queuing are marked BizTalk Message Queuing only.

These tuning parameters are located in the registry. The location of the BizTalk Server registry keys is **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BTSSvc.3.0\Message Queuing**.

MSMQT Adapter Property Schema and Properties

The following table lists the properties in the BizTalk Message Queuing (MSMQT) property schema.

Namespace: **http://schemas.microsoft.com/BizTalk/2003/msmqt-properties**

Property name	Type	Equivalent MSMQT property	Description
ResponseQueue	xs:string	PROPID_M_RESP_QUEUE	Defines the queue where the orchestration would like to r response messages.
InboundResponseQueue	xs:string	PROPID_M_RESP_QUEUE	Defines the location from which any responses should be ser
AdminQueue	xs:string	PROPID_M_ADMIN_QUEUE	Defines the location from which order acknowledgments are
Extension	xs:string	PROPID_M_EXTENSION	Defines an application-specific blob. You cannot use this pro message is a large message.
SenderID	xs:string	None	You should not use this property; use **SourceParty** ins secure identifier (SID) of the user that sends the message.
SourceMachineGuid	xs:string	None	Defines the globally unique identifier (GUID) of the PROPID_ for the incoming message.
MsgID	xs:int	None	Defines the number part of the PROPID_M_MSGID for the message.
Priority	xs:int	PROPID_M_PRIORITY	Defines the message priority for non-transactional incoming only.
AppSpecific	xs:unsignedInt	PROPID_M_APPSPECIFIC	Defines the application-specific integer.
TimeToReachQueue	xs:unsignedInt	PROPID_M_TIME_TO_REACH_QUEUE	Specifies a time-out for the send operation in seconds. BizT generates a delivery exception if a message cannot be s time.
Label	xs:string	PROPID_M_LABEL	Specifies the label for incoming and outgoing messages.
CorrelationId	xs:string	PROPID_M_CORRELATIONID	Defines the application–specific correlation ID, "{GUID}\number" format. Brackets and '\' are required.
IsXactMsg	xs:boolean	None	Set this property to **True** if for incoming transactional messa
IsSecurityIncluded	xs:boolean	None	This property is always **False**.
IsAuthenticated	xs:boolean	None	Set this property to **True** if BizTalk Server authenticated th message.
ArrivedTime	xs:dateTime	PROPID_M_ARRIVEDTIME	Defines the time of arrival for incoming message.

Version	xs:string	None	Defines the version of the Microsoft Message Queuing messa
SentTime	xs:dateTime	PROPID_M_SENTTIME	Defines the time the message was sent.
Class	xs:short	PROPID_M_CLASS	Defines the message class, which is one **System.Messaging.Acknowledgment** values.
HashAlg	xs:unsignedInt	PROPID_M_HASH_ALG	Defines the algorithm used for signing the mess **System.Messaging.HashAlgorithm** for possible values.
Acknowledge	xs:unsignedInt	PROPID_M_ACKNOWLEDGE	Defines whether to request acknowledgment messa **System.Messaging.AcknowledgeTypes** enumeration fo values.
IsFirstInTransaction	xs:boolan	PROPID_M_FIRST_IN_XACT	Set this property to **True** if the message was first in t transaction.
IsLastInTransaction	xs:boolean	PROPID_M_LAST_IN_XACT	Set this property to **True** if the message was last in t transaction.
TransactionId	xs:int	PROPID_M_XACTID	Defines the sender transaction ID.
AuthLevel	xs:boolean	PROPID_M_AUTH_LEVEL	Set this to **True** to force BizTalk Message Queuing to outbound message.
Signature	xs:string	None	Defines the thumbprint of the certificate used to sign the message.
Authenticated	xs:boolean	None	Defines whether the signature of the incoming message was

MSMQT Adapter Deployment and Security Recommendations

The BizTalk Message Queuing adapter is the native Microsoft Message Queuing (also known as MSMQ) adapter in Microsoft BizTalk Server. For more information about the BizTalk Message Queuing adapter, see BizTalk Message Queuing (MSMQT) Adapter .

For more information about the differences between standard Message Queuing and BizTalk Message Queuing, see About the BizTalk Message Queuing Adapter.

Deployment Recommendations for the BizTalk Message Queuing Adapter

- It is not recommended to run both standard Message Queuing and BizTalk Message Queuing on the same computer.
- Even when the BizTalk Message Queuing send and receive locations are running on different servers (different host instances), they must be associated with the same host.

- BizTalk Server does not configure the BizTalk Message Queuing adapter by default. For more information about configuring BizTalk Message Queuing, see Configuring MSMQT Using the Configuration Manager .

- BizTalk Message Queuing level authentication failures do not appear in the event log.

Security Recommendations for the BizTalk Message Queuing Adapter

- Just like other BizTalk Server components, it is recommended you do not put the BizTalk Message Queuing adapter in the perimeter network or Intranet. Unlike standard Message Queuing, which supports both HTTP-based protocol and native protocol, BizTalk Message Queuing supports only native protocol. It is therefore possible to use standard Message Queuing to receive messages in the perimeter network or Intranet by using the HTTP-based protocol and then route the messages to a BizTalk Message Queuing receive location in the processing domain through the native protocol.

- BizTalk Server does not support the use of the BizTalk Message Queuing adapter across Network Address Translation (NAT) firewalls. When you use BizTalk Message Queuing, the firewall acts as a router for the BizTalk Message Queuing traffic. For more information about configuring BizTalk Message Queuing across a firewall, see the Microsoft Web site at http://go.microsoft.com/fwlink/?LinkId=24778.

- You can configure BizTalk Message Queuing to require certificate-based authentication. This occurs at the adapter level, and is different from the party resolution component of a BizTalk pipeline. If configured, the public certificate comes with the inbound message. This is the only client authentication mode available for BizTalk Message Queuing. To use this client authentication mode, you must install BizTalk Message Queuing with Active Directory Integration Mode. When you use this feature, remember to select the **Require Authentication** check box on the property page for the BizTalk Message Queuing receive location.

- When you use the BizTalk Message Queuing adapter, the server running the send and receive locations for this adapter must be behind a firewall that blocks TCP port 1801 and UDP port 3527.

File Adapter

The File adapter transfers files into and out of Microsoft BizTalk Server. The File adapter consists of two adapters—a receive adapter and a send adapter.

This section discusses the workflow and batching support for both the File receive adapter and the File send adapter.

In This Section

- What Is the File Adapter?

- Configuring the File Adapter

- Restrictions When Configuring the File Adapter

- File Adapter Security Recommendations

What Is the File Adapter?

The File adapter consists of two adapters—a receive adapter and a send adapter.

File Receive Adapter

You use the File receive adapter to read messages from files and submit them to the server. The receive adapter reads the file and creates a BizTalk Message object, so that BizTalk Server can process the message. While reading from the file, the adapter locks the file to ensure that no modifications can be made to the file content.

The File receive adapter reads the messages from files on local file systems or on network shares. When the specified location on a network share is unavailable due to network problems, the receive adapter retries the read operation (the number of retries is configurable in the BizTalk Server Administration console). After the message has been read and successfully accepted by the BizTalk Messaging Engine, the receive adapter deletes the file from the file system or network share. If the message was read but the pipeline did not successfully process the message, the adapter puts the message in the suspended queue and then deletes the file from the file system or network share. If the File receive adapter cannot submit or suspend the message to the MessageBox database, it does not delete the original file from the file system or network share.

You can also configure the File receive adapter to rename files when processing them. You should rename files to ensure that the receive adapter does not generate duplicate messages if the receive location is shut down and restarted. This is a configurable option for File receive locations. By default, renaming is disabled. When renaming is enabled, the File receive adapter appends the extension .BTS-WIP to the file. The receive adapter then reads the messages from the renamed file in the receive location and submits it to the server. After the receive adapter has successfully submitted a file, the receive adapter deletes the renamed file from the file system or network share. If a message has been read but failed processing in the pipeline, the receive adapter places the message in the MessageBox database suspended queue, and deletes the renamed file from the network share.

If the File receive adapter successfully read the message but did not successfully store the message in the MessageBox database, the renamed file reverts to its original name, without the .BTS-WIP extension. Note that the receive adapter does not read files with the extension .BTS-WIP if the renaming option is turned on.

File Receive Adapter Batching Support

The File receive adapter submits messages to the server in batches. The File receive adapter starts by building a single batch per receive location by collecting all the readable files available in the receive location. Batches are submitted to the MessageBox database by the receive adapter when all the available files have been collected or when the amount of files collected exceeds the maximum batch size. You can configure the batch size by using BizTalk Explorer.

After all the messages within the batch have been successfully read and submitted into the MessageBox database, the File receive adapter deletes the corresponding files from the receive location. If some of the messages within the batch failed processing, the File receive adapter suspends them and deletes the corresponding files from the receive location. If some or all of the messages fail to be stored in the MessageBox database, the entire batch operation is rolled back and all corresponding files are left unchanged in the receive location.

File Send Adapter

The File send adapter transmits messages from the MessageBox database to a specified destination address (URL). You define the URL, which is a file path and file name, by using wildcard characters related to the message context properties. The File send adapter resolves the wildcard characters to the actual file name before writing the message to the file.

When writing a message to a file, the File send adapter gets the message content from the body part of the BizTalk Message object. The File send adapter ignores other message parts in the BizTalk Message object. After the File adapter writes the message to a file, it deletes the message from the MessageBox database. The File adapter writes files to the file system either directly or by using the file system cache, which can improve performance, particularly for large files.

File Send Adapter Batching Support

The File send adapter gets batches of messages from the MessageBox database and writes them to files in destination locations on the file system or the network share. The size of File send adapter batches is not configurable and is preset to 20. If BizTalk Server fails to write some of the messages within a batch to files, the system resubmits those messages to the MessageBox database for retry processing. You can configure the retry interval and retry count by using the BizTalk Server Administration console.

Configuring the File Adapter

This section describes how to configure the File adapter.

In This Section

- How to Configure a File Receive Handler
- How to Configure a File Receive Location
- How to Configure a File Send Handler
- How to Configure a File Send Port
- File Adapter Property Schema and Properties

How to Configure a File Receive Handler

Use the following procedure to change the host with which the File receive handler is associated.

The user account for the host instance that the File receive handler is running in must have the following permissions:

Permissions required at the file system level

- List Folder / Read Data
- Delete SubFolder and Files

Permissions required at the share level (if accessing a file share)

- Full control

To change the host with which the File receive handler is associated

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then click **Adapters**.
2. In the expanded list of adapters, click **FILE**, in the right pane right-click the receive handler that you want to configure, and then click **Properties**.
3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.
4. Click **OK**.

How to Configure a File Receive Location

You can set File receive location adapter variables either programmatically or by using the BizTalk Server Administration console.

How to Configure a File Receive Location Programmatically

The File adapter stores its configuration information in the SSO database. You can set this configuration information programmatically by using the BizTalk Explorer object model. The BizTalk Explorer object model exposes the **IReceiveLocation** configuration interface that contains the **TransportTypeData** read/write property. This property accepts the File receive location configuration property bag in the form of a name/value pair XML string.

The **TransportTypeData** property of the **IReceiveLocation** interface does not have to be set. If it is not set, default values for the File receive location configuration are used.

The following table lists the configuration properties that you can set programmatically for the File receive location.

Property name	Type	Description	Restrictions	Comments
FileNetFailRetryCount	Long	The number of attempts to access the receive location on the network share if it is temporarily unavailable.	Integer Minimum value: 0 Maximum value: MAX_LONG	If not specified, the default value is set to 5 times.
FileNetFailRetryInterval	Long	The retry interval in minutes between attempts to access the receive location on the network share if it is temporarily unavailable.	Integer Minimumvalue: 0 Maximum value: MAX_LONG	If not specified, the default value is set to 5 minutes.
BatchSize	Long	The number of files this receive location can submit to the server at one time.	Integer Minimum value: 1 Maximum value: 256	If not specified, the default value is set to 20 files.
FileMask	String	The file mask used by the receive location.	String The length of the FilePath and FileMask combined cannot exceed 256 characters.	If not specified, the default value is set to *.xml.
FilePath	String	The path of the folder monitored by the receive location.	String Required The length of the FilePath and FileMask combined cannot exceed 256 characters.	Must be specified
Username	String	User name for account used to access folder.	Min length: 0 Max length: 256	If neither username or password are specified, host credentials are

				used. If null (vt="1") then the value stored in configuration database is used.
Password	String	Password for account used to access folder.	Min length: 0 Max length: 256	If neither username or password are specified, host credentials are used. If null (vt="1") then the value stored in configuration database is used.

The following code shows the format of the XML string you use to set the properties:

How to Configure a File Receive Location with the BizTalk Server Administration Console

To configure the receive location by using the BizTalk Server Administration console, use the following procedure.

To configure receive location variables for a File receive location

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application you want to create a receive location in.

2. In the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

3. In the left pane of the **Receive Port Properties** dialog box, select **Receive Locations**, and in the right pane double-click an existing receive location or click **New** to create a new receive location.

4. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **FILE** from the drop-down list, and then click **Configure** to configure the transport properties for the receive location.

5. In the **FILE Transport Properties** dialog box, do the following:

Use this	To do this
Receive folder	Required. Specify the path to a folder on the file system or network share where the file receive handler reads files. You can enter the path directly in the **Receive folder** text box or select it from the file system by navigating to the folder with the **Browse** button. When browsing for the folder in the **Browse For Folder** dialog box you can also create a new folder by clicking **Make New Folder**. **Type:** String For more information about restrictions on the receive folder property, see Restrictions on the Receive Folder and Destination Location Properties .
File mask	Required. Specify the mask for the files. This mask can contain the standard wildcard value "*". **Default value:** *.xml **Type:** String For information about restrictions on this property, see Restrictions on the File Mask and File Name Properties .
Public address	Specify the public address of this location. BizTalk Server exposes this address to external partners. If this property is not specified, the runtime engine replaces it as: file://<*Receive folder*>/<*File mask*> The value for this property requires an adapter prefix. **Type:** String **Minimum length:** 0 **Maximum length:** 256
Retry count	Specify the number of attempts to access the receive location on a network share if it is temporarily unavailable. **Default value:** 5 **Type:** Long **Minimum value:** 0

	Maximum value: MAX_LONG
Retry interval (min)	Specify the retry interval time (in minutes) between attempts to access the receive location on the network share if it is temporarily unavailable. **Default value:** 5 minutes **Type:** Long **Minimum value:** 0 **Maximum value:** MAX_LONG

6. On the **General** tab, click **Advanced Settings**, and in the **Advanced Settings** dialog box, do the following:

Use this	To do this
Rename files while reading	Specify whether to rename files before picking them up for processing. For more information about using this option, see File Adapter **Default Value:** False **Type:** Boolean
Polling interval (ms)	Specify the interval in milliseconds that the File adapter will poll the specified location for new files. **Default Value:** 60,000 **Type:** Int **Minimum value:** 1000 (set to 1 to disable polling) **Maximum value:** 3600000
Retry count	Specify the number of times that the File adapter will attempt to delete a file that it has read and submitted to BizTalk Server. **Default Value:** 5 **Type:** Int **Minimum value:** 0 **Maximum value:** 100
Retry interval	Specify the initial interval in milliseconds that the File adapter waits before

(ms)	attempting to delete a file that it has read and submitted to BizTalk Server. This interval will double after each retry interval up to the specified maximum retry interval value. **Default Value:** 10 **Type:** Int **Minimum value:** 1 **Maximum value:** 1000
Maximum retry interval (ms)	Specify the maximum retry interval time in milliseconds that the File adapter waits before attempting to delete a file that it has read and submitted to BizTalk Server. **Default Value:** 300000 **Type:** Int **Minimum value:** 1000 **Maximum value:** 900000

7. Click **OK**.

8. On the **Authentication** tab, in the **File Transport Properties** dialog box, do the following:

Use this	To do this
Use these credentials when host does not have access to network share	Specify to use alternative credentials when the host instance for the File adapter does not have the necessary rights to a network share. This option is only valid when accessing a network share. **Default Value:** False **Type:** Boolean
User name	Specify the user name that has access to the network share. **Type:** String
Password	Specify the password for the account that has access to the network share.

	Type: String

9. In the **File Transport Properties** dialog box, on the **Batching** tab, do the following:

Use this	To do this
Number of messages in a batch	Specify the maximum number of messages to be submitted in a batch. **Default Value:** 5 **Type:** Int **Minimum value:** 1 **Maximum value:** 256
Maximum batch size (in bytes)	Specify the maximum total bytes for a batch. **Default Value:** 102400 **Type:** Int **Minimum value:** 1024 **Maximum value:** MAX_LONG

10. The File adapter will limit the batch to whichever value is reached first, maximum number of messages or maximum allowed bytes.

11. Click **OK**.

12. Enter the appropriate values in the **Receive Location Properties** dialog box to complete the configuration of the receive location and click **OK** to save settings. For information about the **Receive Locations Properties** dialog box, see How to Create a Receive Location.

How to Configure a File Send Handler

File send handler is the host instance on which the File send adapter runs. This topic explains how to change the default configuration of the File send handler.

The BTSNTSvc.exe service hosts the File send handler. For the File send handler to send messages to a specified location, this service needs to have read/write privileges for the specified folder on the file system or network share.

Use the following procedure to change the host with which the File send handler is associated.

To configure a File send handler

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **FILE**, in the right pane right-click the send handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the send handler will be associated.

4. Click **OK**.

How to Configure a File Send Port

You can set File send port adapter variables either programmatically or by using the BizTalk Server Administration console.

How to Configure the Send Port Programmatically

The File adapter stores its configuration information in the SSO database. Configuration information is stored in a custom XML property bag. The File send handler property schema, bts_file_properties.xsd, defines the File adapter-specific properties. You use these properties to configure the File send ports, as well as for passing adapter-specific information within the server.

You can configure the send ports programmatically by using the BizTalk Explorer object model.The BizTalk Explorer object model exposes the **ITransportInfo** adapter configuration interface for File send ports, which contains the **TransportTypeData** read/write property. This property accepts the File send handler configuration property bag as a name/value pair XML string.

Setting the **TransportTypeData** property of the **ITransportInfo** interface is not required. If it is not set, the default values for the File send handler configuration are used.

The following table lists the configuration properties you can set programmatically in the BizTalk Explorer object model for the File send handler location.

Property name	Type	Description
CopyMode	Long	Define the copy mode to use when writing a message to a file. Valid values are: **Append (0).** The File send handler opens a file if it exists and appends a message to the end of the file. If the file does not exist, the File send handler creates a new file. **Create new (1).** If the file does not exist, the File send handler creates a new file and writes to it. If the file already

		exists, the File send handler reports an error and then follows common adapter retry logic for send ports. This is a default copy mode for the File send handler. **Overwrite (2).** The File send handler opens a file if it exists and overwrites its content. If the file does not exist, the File send handler creates a new file.
AllowCacheOnWrite	Boolean	Defines whether the File adapter uses file system caching when writing messages to a file.

If any of the configuration properties do not have a value on the message context, the File send handler uses its default value.

You can set configuration properties programmatically on a message context. You can set these properties in an orchestration or in a custom pipeline component. The following rules apply when using these properties:

- If the configuration property is set in an orchestration or in a custom pipeline component in a receive pipeline, then:
  - If a message is sent to a static send port, the property value will be overwritten with the value configured for that send port.
  - If a message is sent to a dynamic send port, the property value will not be overwritten.
- If a configuration property is set in a custom pipeline component in a send pipeline, then:
  - The value will not be overwritten regardless of whether the message is sent to a static or dynamic send port.

The following code shows the format of the XML string you can use to set the properties:

How to Set the File Adapter Configuration Properties for a Dynamic Send Port

A dynamic send port does not provide any transport configuration options in BizTalk Explorer because it is expected that these properties will be provided with the context properties associated with the message. These properties can be set in a custom pipeline or in an orchestration. To set message configuration properties in an orchestration you can do the following:

- Add a **Construct Message** shape to your orchestration.
- Configure the **Construct Message** shape to construct a new message. (for example Message_2)

- Add a **Message Assignment** shape to the **Construct Message** shape.

- Add code to the **Message Assignment** shape to initialize the message that you constructed and to set the appropriate configuration properties for the message. The following code initializes a message named Message_2 that was constructed with a **Construct Message** shape and then sets two configuration properties for the message. In this scenario, Message_1 was originally received by the orchestration:

How to Configure the Send Port with the BizTalk Server Administration Console

To configure the send port by using the BizTalk Server Administration console, use the following procedure.

To configure variables for a File send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. See How to Create a Send Port for more information. Configure all of the send port options and specify **FILE** for the **Type** option in the **Transport** section of the **General** tab.

2. On the **General** tab, in the **Transport** section, click the **Configure** button next to **Type**.

3. In the **File Transport Properties** dialog box, on the **General** tab do the following:

Use this	To do this
Destination Location	Specify the path to the location on the file system or public share to write the output messages. You can enter the path directly in the **Destination Location** text box or select it from the file system by navigating to the folder with the **Browse** button. When browsing for the folder in the **Browse For Folder** dialog box you can also create a new folder by clicking **Make New Folder**. **Type:** String
File name	Specify the name of the file where the File send handler writes the message. For information about restrictions on this property, see Restrictions on the File Mask and File Name Properties . For information about using macros in the file name, see Restrictions on Using Macros in File Names .
Copy mode	Define the copy mode to use when writing a message to a file. Valid values are: **Append.** The File send handler opens a file if it exists and appends a message to the end of the file. If the file does not exist, the File send handler creates a

	new file. **Overwrite**. The File send handler opens a file if it exists and overwrites its content. If the file does not exist, the File send handler creates a new file. **Create new**. If a file does not exist, the File send handler creates a new file and writes to it. If the file already exists, the File send handler reports an error and then follows common adapter retry logic for send ports. This is a default copy mode for the File send handler.
Allow Cache on write	Define whether to use file system caching when writing a message to a file. Valid options are: **False.** Do not use the file system cache. **True.** Use the file system cache. **Default Value:** False

4. On the **Authentication** tab, in the **File Transport Properties** dialog box, do the following:

Use this	To do this
Use these credentials when host does not have access to network share	Specify to use alternative credentials when the host instance for the File adapter does not have the necessary rights to a network share. This option is only valid when accessing a network share. **Default Value:** False **Type:** Boolean
User name	Specify the user name that has access to the network share. **Type:** String
Password	Specify the password for the account that has access to the network share. **Type:** String

5. Click **OK** and **OK** again to save settings.

File Adapter Property Schema and Properties

The following table contains the properties in the File adapter property schema.

Namespace: http://schemas.microsoft.com/BizTalk/2003/file-properties

Name	Type	Description
CopyMode	xs:unsignedInt	Defines the copy mode to use when writing a message to a file. Valid values are: **Append (0).** The File send handler opens a file if it exists and appends a message to the end of the file. If the file does not exist, the File send handler creates a new file. **Create new (1).** If a file does not exist, the File send handler creates a new file and writes to it. If the file already exists, the File send handler reports an error and then follows common adapter retry logic for send ports. This is a default copy mode for the File send handler. **Overwrite (2).** The File send handler opens a file if it exists and overwrites its content. If the file does not exist, the File send handler creates a new file.
AllowCacheOnWrite	xs:Boolean	Defines whether the File adapter uses file system caching when writing messages to a file.
ReceivedFileName	xs:string	Defines the full name of the file from which the File adapter reads the message.
FileCreationTime	xs:datetime	Defines the time that the file was written to the folder that is monitored by the File receive adapter.
Username	xs:string	Defines the user name for the account used when specifying alternative credentials to access a network share.
Password	xs:string	Defines the password for the account used when specifying alternative credentials to access a network share.

Restrictions When Configuring the File Adapter

This section contains information that is useful when configuring the File adapter.

In This Section

- Restrictions on the File Mask and File Name Properties
- Restrictions on Using Macros in File Names
- Restrictions on the Receive Folder and Destination Location Properties

Restrictions on the File Mask and File Name Properties

The file mask is a string that specifies the type of file that the File receive handler will pick up from the receive location. The file name is a string that specifies the name of the file where the File send handler will write the message.

The following restrictions apply to the file name and file mask properties:

- Only one file mask or file name can be specified per receive location or send port.
- The full path or part of the path along with the file mask or file name is not allowed. The file mask and file name always represent a name without the path.
- The file mask and file name are not case-sensitive.
- The file name cannot contain any of the following characters: < > : / | " ? * ;
- The file mask cannot contain any of the following characters: < > : / | " ;
- The following reserved device names cannot be used as the name of a file: CON, PRN, AUX, CLOCK$, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9. In addition, any combinations of these with extensions are not allowed.
- The File adapter considers only the first three characters in a file mask extension when matching files in receive locations with a mask. For example, if a file mask is "*.xml", files with the extension ".xml[any other symbol]" will be picked up. In addition, if a file mask is "*.xmln", files with the extension ".xml[any other symbol]" will be picked up.
- The total length of the file path, file mask, and file name (without macro substitution) must not exceed 256 characters. (This is a restriction of the MessageBox database.)
- The file path cannot begin with "\\?".
- Mapped network drive letters cannot be used in the file path, because they are user session based.

The BizTalk Messaging Engine always validates the file name and file mask properties at design time by using the items previously listed. In addition, the File adapter validates the file name and file mask properties at run time if the adapter sends the message on a dynamic port.

Restrictions on Using Macros in File Names

You can use a predefined set of macros to dynamically create the files in which the File send handler writes messages. Before creating a file on the file system, the File send handler replaces all the macros in the file name with their individual values. You can use several different macros in one file name.

You can use the file name macros while configuring the File send handler in BizTalk Explorer, or by using the BizTalk Explorer object model.

The File send handler does not replace the macros with a value if any of the following are true:

- The corresponding system property is not set.
- The macro is misspelled.
- The value for the macro contains symbols that are not valid in the file name.

If any of these conditions occur, the File send handler leaves the macros in the file name untouched, for example Myfile_%MessageID%.xml.

The following table lists the supported macros and describes how the File send handler replaces them.

Macro name	Substitute value
%datetime%	Coordinated Universal Time (UTC) date time in the format YYYY-MM-DDThhmmss (for example, 1997-07-12T103508).
%datetime_bts2000%	UTC date time in the format YYYYMMDDhhmmsss, where sss means seconds and milliseconds (for example, 199707121035234 means 1997/07/12, 10:35:23 and 400 milliseconds).
%datetime.tz%	Local date time plus time zone from GMT in the format YYYY-MM-DDThhmmssTZD, (for example, 1997-07-12T103508+800).
%DestinationParty%	Name of the destination party. The value comes from the message context property **BTS.DestinationParty**.
%DestinationPartyQualifier%	Qualifier of the destination party. The value comes from the message context property **BTS.DestinationPartyQualifier**.
%MessageID%	Globally unique identifier (GUID) of the message in BizTalk Server. The value comes directly from the message context property **BTS.MessageID**.
%SourceFileName%	Name of the file from which the File adapter read the message. The file name includes the extension and excludes the file path,

	for example, Sample.xml. When substituting this property, the File adapter extracts the file name from the absolute file path stored in the **FILE.ReceivedFileName** context property. If the context property does not have a value—for example, if a message was received on an adapter other than the File adapter—the macro will not be substituted and will remain in the file name as is (for example, C:\Drop\%SourceFileName%).
%SourceParty%	Name of the source party from which the File adapter received the message.
%SourcePartyQualifier%	Qualifier of the source party from which the File adapter received the message.
%time%	UTC time in the format hhmmss.
%time.tz%	Local time plus time zone from GMT in the format hhmmssTZD (for example, 124525+530).

Restrictions on the Receive Folder and Destination Location Properties

The file receive location is a string that contains a path to a folder on a file system or network share from which the File receive handler reads files. The file destination location is a string that contains a path to a folder on a file system or network share where the File send handler writes files.

The following restrictions apply to the receive folder and destination location properties:

- Existence of the file path on a file system or network share is not required at the time when you specify the property in the user interface or in the BizTalk Explorer object model.
- The file path must always be absolute.
- You can specify the file path by using Universal Naming Convention (UNC) format (for example, \\<*server*>\<*share*>).
- If the file path is in UNC format, the server name must not contain the following characters: ` ~ ! @ # $ ^ & * ( ) = + [ ] { } \ | ; : ' " , < > / ? ;
- You cannot use parent (\..\) and current (\.\) folder symbols in any part of the file path.
- The file path is not case-sensitive.
- The file path cannot contain any of the following characters: < > : / | " ? * ;

- You cannot use the following reserved device names in the file path: CON, PRN, AUX, CLOCK$, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9.

- Total length of file path, file mask, or file name (without macro substitution) must not exceed 256 characters. (The MessageBox database imposes this restriction.)

- The File adapter does not support Unicode specification of the file path (for example, "\\?\").

Restrictions on the receive folder property only:

- Do not set the receive folder property to a folder that uses the Microsoft Windows NT Distributed File System with a symbolic link. If you are using Windows NT Distributed File System, you can only use folders with straight network paths in File adapter receive locations.

- When documents are sent to a UNC path, and you have more than one server receiving documents at the receive location for the File adapter, only one server will pick up and process most of the documents sent to that UNC path. For more information about file renaming, see the File Receive Adapter section of File Adapter .

The File adapter validates the file path at design time by using the previously mentioned rules. In addition, the File adapter validates the message at run time if the adapter sends the message through a dynamic port with a File adapter.

File Adapter Security Recommendations

The File adapter transfers files into and out of BizTalk Server from and to a directory. It is recommended you follow these guidelines for securing and deploying the File adapter in your environment:

- It is recommended that you do not open ports to connect to a file share in the perimeter network. You should use the File adapter in environments where there is a high level of trust, such as an intranet. It is not recommended to use the File adapter for receiving and sending messages across the Internet.

- It is recommended that you set strong discretionary access control lists (DACL) in the receive location's drop directories. For example, you must set read, write, delete files, and delete subfolders and files permissions to the directory from which the file receive location picks up messages, so that only authorized users can drop messages in this location.

- When you use the File adapter to pick up critical data, it is recommended to use Internet Protocol Security (IPSec.)

FTP Adapter

The FTP adapter exchanges data between an FTP server and Microsoft BizTalk Server, and allows for the integration of vital data stored on a variety of platforms with line-of-business applications.

The FTP adapter can transfer a large number of files from an FTP server to BizTalk Server. It can also transfer files as part of an orchestration.

The FTP adapter consists of two adapters—a receive adapter and a send adapter.

In This Section

- What Is the FTP Adapter?
- Configuring the FTP Adapter
- FTP Adapter Security Recommendations

What Is the FTP Adapter?

This section describes the FTP receive and send adapters, as well as security and best-practice information.

FTP Receive Adapter

The FTP receive adapter enables you to move data from an FTP server to BizTalk Server.

Key features of the FTP receive adapter are:

- Pulling files from the FTP server on demand
- Running polls based on a configurable schedule
- Polling the FTP server and sending data directly to BizTalk Server
- Specifying the FTP server as an IP address, port, password, and host name
- Guaranteed file delivery

The FTP receive adapter also works with the BizTalk Server Administration console and BizTalk Explorer to configure and administer each receive function, which is composed of the following configuration items:

- Poll interval to run an FTP command (for example, 60 minutes)
- Information with which to route the document to a specific BizTalk send port or receive location

FTP Send Adapter

The FTP send adapter enables you to move data from BizTalk Server to an FTP server.

Key features of the FTP send adapter are:

- Ability to run sends on demand
- Guaranteed delivery

FTP Adapter Supported Platforms

Using the FTP adapter, you can access information stored in an FTP server on any of the following platforms: Solaris 9.0, HP-UX, LINUX (Redhat 7.x), IBM O/S 390 running MVS, AS400 OS/400 V5R1, Microsoft Windows 2000 Server Service Pack 4, Microsoft Windows 2000 Advanced Server Service Pack 4, Microsoft Windows Server 2003.

In This Section

- FTP Adapter Security
- Best Practices for the FTP Adapter

FTP Adapter Security

The first step in securing your data is securing your server and limiting access to the data. You can ensure that the FTP server is secure by using a dedicated connection and limiting the server and the connection between BizTalk Server and the FTP host. Use a dedicated connection and limit the number of people who have access to the computers. FTP is not a secure protocol, so it will always be vulnerable, but using a dedicated connection and locking down servers are the best ways to keep the environment safe.

The unsecured nature of FTP requires that you carefully consider the safety of your data. For example, the FTP protocol sends user IDs and passwords as plain text to log on to the receiving FTP server; therefore you should use FTP only when sending and receiving files from trusted partners over a dedicated connection.

When using an FTP send port, you must specify and store a user ID and password combination when configuring the send port. The adapter uses this information to connect to the FTP server. The user credentials are stored in a SQL Server database in plain text. In a dynamic send port, credentials are sent to the FTP server. If production environment requirements warrant stronger security, use anonymous credentials to the server.

Additional security tips:

- For security reasons, when the system prompts you for an account, it is recommended that you enter an existing user account, and not the local system account. This enables you to implement better security, and allows the adapter to run in unattended mode, without logging on.

- The system sends FTP protocol user IDs and passwords as plain text. When transferring files through FTP, the credentials are exposed. The expected user scenario for this adapter is over a secure line.

Best Practices for the FTP Adapter

The following are recommended best practices:

- Delete partially received files from the temporary folder on a regular basis to keep files from using computer resources and potentially disrupting service.

- When working with a streaming server, deny read access to the new file until the MessageBox database receives the entire file. If a partial file is submitted to the MessageBox database by the FTP adapter, the MessageBox database will successfully store the message, but the FTP adapter will not be able to delete the partial message from the receive location.

- To ensure high availability for the FTP adapter receive handler, the FTP adapter receive handler should be configured to run in a clustered BizTalk Host instance. For more information see Considerations for Running Adapter Handlers within a Clustered Host.

Configuring the FTP Adapter

This section describes how to configure an FTP adapter.

In This Section

- How to Configure an FTP Receive Handler
- How to Configure an FTP Receive Location
- Configuring a Receive Location to Use the FTP Transport
- How to Configure an FTP Send Handler
- How to Configure an FTP Send Port
- Configuring an FTP Adapter to Work with Legacy Hosts
- FTP Adapter Property Schema and Properties

How to Configure an FTP Receive Handler

You can set FTP receive handler properties in the BizTalk Server Administration console. You use these receive handler properties as the receive location configuration values if properties are not set on the individual FTP receive locations.

To configure an FTP receive handler

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

3. In the expanded adapter list, select **FTP**.

4. In the right pane, double-click the receive handler.

5. In the **Adapter Handler Properties** dialog box, in the **Host name** drop-down list, select the host with which the receive handler will be associated, and then click **Properties**.

6. In the **FTP Transport Properties** dialog box, do the following.

Use this	To do this
Maximum Files	Specify the maximum number of files per BizTalk Server batch. Zero (0) indicates no limit. **Default value:** 0
Maximum Size	Specify the maximum number of bytes per BizTalk Server batch. Zero (0) indicates no limit. **Default value:** 0
Address	Specify the address of the firewall, either DNS name or IP address.
Mode	Specify the mode in which the adapter connects to the FTP server. **Valid values:** Passive or Active **Default Value:** Active
Password	Specify the password for the firewall.
Port	Specify the port for the firewall. **Valid values:** 1 through 65535 inclusive **Default value:** 21

Type	Specify the type of firewall deployed. **Valid values:** Socks 4, Socks 5, None **Default value:** None
User	Specify the user name for the firewall.
Account	Specify the account name for the FTP server. This option is deprecated in BizTalk Server 2006 and use of this property is discouraged.
After GET	Specify the FTP commands to run after the file GET. Separate commands with a semicolon (;).
Before GET	Specify the FTP commands to run before the file GET. Separate commands with a semicolon (;).
Error Threshold	Specify the number of errors that BizTalk Server can encounter before the location is disabled. **Default value:** 10
Log	Specify the location to save a copy of the log file. Use this file to diagnose error conditions when sending or receiving files through FTP.
Max File Size	Specify the maximum downloadable file size, in megabytes (MB). Zero (0) indicates no limit. **Default Value:** 100
Password	Specify the user password to log on to the FTP server.
Representation	Select how FTP receives the data. **Valid Values:** binary or ASCII **Default Value:** binary
User Name	Specify the user name to log on to the FTP server.
Temporary Folder	Specify the location for a temporary folder. Use this folder to ensure recovery from a transfer failure.

7. Click **OK** and click **OK** again.

How to Configure an FTP Send Port

You can set FTP send port adapter variables in the BizTalk Server Administration console. If properties are not set for the send port, the default send handler values set in the BizTalk Server Administration console are used.

To configure variables for an FTP send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. for more information. Configure all of the send port options, and in the **Transport** section of the **General** page, specify **FTP** for the **Type** option.

2. On the **General** page, in the **Transport** section, click the **Configure** button next to **Type**.

3. In the **FTP Transport Properties** dialog box, do the following:

Use this	To do this
Address	Specify the address of the firewall, either a DNS name or an IP address.
Mode	Select the mode in which the adapter connects to the FTP server. **Valid values:** Passive and Active **Default value:** Active
Password	Specify the password for the firewall.
Port	Specify the port for the firewall. **Valid values:** 1 through 65535 inclusively **Default value:** 21
Type	Select the type of firewall deployed. **Valid values:** Socks 4, Socks 5, None **Default value:** None
User	Specify the user name for the firewall.
Account	Optional. Specify the account name for the FTP server. This option is deprecated in BizTalk Server 2006 and use of this property is discouraged.
After Put	Specify the FTP commands to run after the file PUT. Separate commands with a semicolon (;).

Allocate Storage	Specify whether to allocate storage space for legacy host systems. This option is provided for backward compatibility but is not used by BizTalk Server 2006. **Valid values:** No and Yes **Default value:** No
Before Put	Specify the FTP commands to run before the file PUT, such as commands to change default values on the FTP server. Separate commands with a semicolon (;). No open command is required.
Folder	Specify the location to move the files to on the FTP server.
Log	Specify the location to save a copy of a log file. Use this file to diagnose error conditions when sending or receiving files through FTP.
Password	Specify the user password to log on to the FTP server.
Port	Specify the port address for this FTP server. **Default value:** 21
Representation	Select how FTP sends the data, either binary or ASCII. **Valid values:** binary or ASCII **Default value:** binary
Server	Specify the server name or IP address of the FTP server.
SSO Affiliate	Specify the Enterprise Single Sign-On affiliate application.
Target File Name	Specify an alternative name for the file. Retaining the default name will guarantee unique message names for each message sent. **Default value:** %MessageID%.xml
User Name	Specify the user name to log on to the FTP server.
Connection Limit	Specify the maximum number of concurrent FTP connections that can be opened to the server. A value of 0 means no limit. **Default value:** 0
Temporary Folder	Specify the location for a temporary folder on the FTP server. You use this to ensure recovery from a transfer failure.

4. Click **OK** and **OK** again to save settings.

Configuring an FTP Adapter to Work with Legacy Hosts

This topic addresses what you need to know to facilitate communication between the FTP adapter and a mainframe computer. For more information, review the documentation for your specific operating system.

MVS

To send files to an FTP server on a mainframe, the mainframe must support IBM Generation Data Group (GDG). In the name field, each file name will append a (+1) to the destination file name (a full path with quotes around it).

AS400

There are three methods of naming files and defining their paths when transferring files to and from an AS400 system:

- **Filename field.** When sending a file to an FTP server, enter the file name in the Filename field. The file name must conform to the file-naming conventions of the AS400 system because the file will be stored in the Library File System.

- **Quote command.** You use the Quote command to run a script on the remote computer. You enter the Quote command into the Before GET, Before PUT, After GET, and After PUT fields on any of the endpoints. Enter the Quote command in the following format:

- **Integrated File System (IFS).** IFS is an area on the AS400 system that allows the storage of PC-based files and therefore the same naming conventions as a PC. To use the IFS instead of the default Library File System, the first command to use is `quote site namefmt 1`. This command tells the AS400 system to use the IFS naming conventi

FTP Adapter Property Schema and Properties

The following table contains the properties in the FTP adapter property schema.

Namespace: http://schemas.microsoft.com/BizTalk/2003/ftp-properties

Name	Type	Description
RepresentationType	xs:string	Specifies how the FTP adapter sends data. **Valid values:** binary or ASCII
SSOAffiliateApplication	xs:string	Specifies the Enterprise Single Sign-On affiliate application to use on the FTP send port.

UserName	xs:string	Specifies the user name to log on to the FTP server when sending messages.
Password	xs:string	Specifies the password to use when logging on to the FTP server when sending messages.
BeforePut	xs:string	Specifies the FTP commands to run before the file PUT, such as commands to change default values on the FTP server. Separate commands with a semicolon (;). No open command is required.
AfterPut	xs:string	Specifies the FTP commands to run after the file PUT. Separate commands with a semicolon (;).
ReceivedFileName	xs:string	Specifies the full name of the file from which the FTP adapter reads the message.
MaxConnections	xs:unsignedInt	Specifies the maximum number of concurrent FTP connections that can be opened to the server. A value of 0 means no limit.
CommandLogFileName	xs:string	Specifies the location to save a copy of a log file that can be used to diagnose error conditions when sending or receiving files through FTP.
AllocateStorage	xs:boolean	This option is deprecated in BizTalk Server 2006 and use of this property is discouraged.
PassiveMode	xs:boolean	Specifies the mode in which the adapter connects to the FTP server. If **PassiveMode** is false then the adapter connects to the FTP server using Active mode. The default value for this property is false.
SpoolingFolder	xs:string	Specifies the location for a temporary folder on the FTP server. You use this to ensure recovery from a transfer failure.

FTP Adapter Security Recommendations

With the FTP adapter, BizTalk Server can receive files from a File Transfer Protocol (FTP) server and send files to an FTP server for other applications. BizTalk Server does not act as an FTP server.

FTP is, by nature, not secure: The user name, password, and other credentials traverse the network in clear text. Likewise, files uploaded or downloaded move across in clear text and can be easily viewed or tampered with along the way. Moreover, an attacker could spoof the FTP server itself (known as a rogue server attack), in which case there is no way to tell if a particular FTP server is indeed the computer with which you intended to communicate. Therefore, it is risky to use the FTP adapter for sensitive data over a network that is not

secure unless you can assure security at the file or messaging layers through encryption and digital signatures.

For general security considerations when you use the FTP protocol, see the Internet FAQ Archives Web site at http://go.microsoft.com/fwlink/?LinkId=24779. For general security recommendations when you use the FTP protocol and firewalls, see the ISAserver.org Web site at http://go.microsoft.com/fwlink/?LinkId=25225. For more information about the FTP adapter, see FTP Adapter.

It is recommended that you use the following guidelines for securing and deploying the FTP adapter in your environment:

- BizTalk Server does not configure the FTP adapter by default. For more information about configuring the FTP adapter, see Configuring the FTP Adapter.
- The FTP adapter supports FTP Request for Comments (RFC) 959. For more information about FTP RFC 959, see the World Wide Web Consortium (W3C) Web site at http://go.microsoft.com/fwlink/?LinkId=24781. The FTP adapter does not support the Secure FTP (SFTP) protocol.
- You can use the FTP adapter across firewalls. Depending on the type of firewall you use, you may need to configure one or more of the following firewall properties: username, password, computer, port, firewall type (none, socks 4, socks 5), and mode.
- It is recommended you place the remote FTP server in a secure location. You must ensure the physical and network security of this server to minimize rogue server attacks.
- The FTP adapter supports the use of Enterprise Single Sign-On (SSO). For more information, see Enterprise Single Sign-On .
- The user context the FTP adapter uses to connect to the FTP server must have write permissions in the FTP server, because the adapter removes the file from the server.

HTTP Adapter

You use the HTTP adapter to exchange information between Microsoft BizTalk Server and an application by means of the HTTP protocol. HTTP is the primary protocol for interbusiness message exchange. Applications can send messages to a server by sending HTTP POST or HTTP GET requests to a specified HTTP URL. The HTTP adapter receives the HTTP requests and submits them to BizTalk Server for processing. Similarly, BizTalk Server can transmit messages to remote applications by sending HTTP POST requests to a specified HTTP URL.

The HTTP adapter consists of two adapters—a receive adapter and a send adapter. The HTTP receive adapter is a Microsoft Internet Information Services (IIS) Internet Server Application Programming Interface (ISAPI) extension that the IIS process hosts, and controls the receive locations that use the HTTP adapter. The HTTP send adapter controls the send ports that use the HTTP adapter.

This section discusses the workflow and batching support for both the HTTP receive adapter and the HTTP send adapter.

In This Section

- HTTP Receive Adapter
- HTTP Send Adapter
- Configuring the HTTP Adapter
- HTTP Adapter Security Recommendations

HTTP Receive Adapter

Microsoft Internet Information Services (IIS) hosts the HTTP receive adapter and accepts HTTP requests that contain messages. The receive location for the HTTP receive adapter is a distinct URL configured through the BizTalk Server Administration console or BizTalk Explorer.

You can configure the HTTP receive adapter for either asynchronous submission or synchronous submission from the client. Asynchronous submissions are one-way submissions and synchronous submissions are two way or request-response submissions.

You use IIS security for authentication and authorization of incoming requests.

HTTP GET and HTTP POST Requests

The HTTP receive adapter can receive messages in two different ways—by an HTTP POST request or an HTTP GET request.

When an HTTP receive adapter gets messages on an HTTP POST request, the following sequence of events occurs:

1. The URL configured in BizTalk Server receives a new message on the receive location.
2. The receive adapter creates a BizTalk Message object so that the message can be submitted to the server.
3. The receive adapter creates the BizTalk Message object with only one part—the body part.
4. After the message has been read and successfully submitted to the server, the HTTP receive adapter sends an HTTP code 202 back to the client indicating that the request was accepted.

   Optionally, the HTTP receive adapter can send a message correlation token on the HTTP response. This correlation token represents the message within BizTalk Server. If the

HTTP receive location is in a request-response port, the adapter returns success code 200 along with the response message.

When an HTTP receive adapter processes messages from an HTTP GET request, the receive adapter creates a BizTalk Message object and puts the decoded query string of the HTTP GET request into the BizTalk message body part. The HTTP adapter selects the query string that is placed into the BizTalk message body part using the following algorithm:

- If the HTTP receive adapter receives an HTTP GET request, it splits the incoming URI string into two parts, using the question mark (?) symbol as a delimiter.

- The first part of the URI string, the part before the question mark delimiter, is copied into the **InboundTransportLocation** property on the message context. The **InboundTransportLocation** property uniquely identifies the location where BizTalk Server received the message. The engine uses this property to determine which receive location to run for the message.

- The HTTP adapter takes the rest of the URI string, the part after the question mark delimiter, and decodes and copies it into the BizTalk message body part.

- If an empty HTTP GET or HTTP POST operation is received by the HTTP receive adapter, it is rejected.

HTTP Receive Adapter Processing of a GET Request

The following are examples of how the HTTP receive adapter processes messages received by HTTP GET requests. These examples assume that the HTTP receive adapter is configured with the following two receive locations:

1. Given the following HTTP GET request for the client:

2. The action taken by the HTTP receive adapter is as follows:

   Set the **InboundTransportLocation** property on the message context equal to /vroot/BTSHTTPReceive.dll, and the BizTalk Message object body part equal to LocationID=1.

3. Given the following HTTP GET request for the client:

4. The action taken by the HTTP receive adapter is as follows:

   Set the **InboundTransportLocation** property equal to /vroot/BTSHTTPReceive.dll, and the BizTalk Message object body part equal to LocationID=1&MyParam=My Value.

5. Given the following HTTP GET request for the client:

6. The action taken by the HTTP receive adapter is+ as follows:

   Reject the request due to incorrect formatting of the HTTP GET request.

Batching Support for the HTTP Receive Adapter

The HTTP receive adapter submits messages to the server in batches. The size of the batch used to submit messages to the server can be configured on the HTTP adapter receive handler.

HTTP Receive Adapter Support for Suspending Failed Requests

The BizTalk Server 2006 HTTP receive adapter has a configuration setting, **Suspend Failed Requests**, to control what happens with an HTTP request if it fails inbound processing due to a receive pipeline failure, a mapping failure, or a routing failure. The setting has two possible values:

- **False.** This is the default setting. The HTTP receive adapter discards messages that fail inbound processing due to a receive pipeline failure, a mapping failure, or a routing failure. Additionally, an error status code 401 or 500 is sent to the client. This is the same behavior as the HTTP receive adapter in BizTalk Server 2004.

- **True.** The HTTP receive adapter suspends messages that fail inbound processing due to a receive pipeline failure, a mapping failure, or a routing failure. For one-way receive ports an **Accepted** status code 202 is sent to the client. For two-way receive ports an **Error** status code 500 is sent to the client.

Chunked Encoding Support for the HTTP Receive Adapter

The HTTP receive adapter accepts HTTP requests with chunked encoded body messages. The receive adapter uses chunked encoding to send response messages when the body size is larger than 4 KB. Chunked encoding can be turned off by setting the DWORD registry entry described in HTTP Adapter Configuration and Tuning Parameters

Client Certificates for the HTTP Receive Adapter

Whenever a secure connection with a client certificate is used for the HTTP receive location, the HTTP receive adapter obtains the client certificate thumbprint from Microsoft Internet Information Services (IIS) and adds it to the message context of all messages that were received over HTTPS on that location. The HTTP receive adapter sets the following system properties:

Status Codes Returned by the HTTP Receive Adapter

The following list contains status codes returned by the HTTP receive adapter.

- **200 OK.** The adapter successfully processed the request message and generated a response. The adapter returns this status code on the HTTP response from the HTTP request-response port.

- **202 Message Accepted.** The adapter successfully submitted the message into the server or a one-way request is suspended. The adapter returns this status code on the HTTP response from a one way HTTP receive port.

- **401 Access Denied.** The HTTP request is received on an authentication-required receive port and the security check for that message failed. For example, the party was not resolved or the message was not decrypted.

- **500 Internal Server Error.** A general failure to process the HTTP request. The message is not suspended by BizTalk Server unless the configuration setting **Suspend Failed Requests** is set to **True** for a two-way receive port

HTTP Send Adapter

The HTTP send adapter gets messages from BizTalk Server and sends them to a destination URL on an HTTP POST request. The HTTP send adapter gets the message content from the body part of the BizTalk Message object. The HTTP send adapter ignores all other parts of the BizTalk Message object.

After the adapter sends the message to a destination URL and the BizTalk Messaging Engine receives the HTTP success status code, the HTTP send adapter deletes the message from the MessageBox database.

Redirection of HTTP messages is supported and can be configured on the send port.

BizTalk Server hosts the HTTP send adapter as a native BizTalk application. It supports one-way sending of messages as well a solicit-response transmission. The send location for the HTTP send adapter is a distinct URL that you configure through the send port. This unique URL can include query strings appended to the base URL.

Batching Support for the HTTP Send Adapter

The HTTP send adapter does not support batching operations.

Chunked Encoding Support for the HTTP Send Adapter

If the **Enable chunked encoding** configuration option is enabled, then the HTTP send adapter sends request messages using chunked encoding if the request size exceeds 8 KB. If the HTTP proxy server is used, the HTTP send adapter does not use chunked encoding and always stages the data before sending. The **Enable chunked encoding** configuration option is enabled by default.

When the send adapter receives a response message, it can accept response messages with a chunked encoded body part.

Client Authentication for the HTTP Send Adapter

The HTTP send adapter authenticates with the destination server by using one of the following authentication types:

- **Anonymous.** The HTTP adapter does not send any credentials when connecting to the destination server. If the destination server permits anonymous authentication then the credentials of the configured anonymous account on the destination server are used.

- **Basic.** The HTTP adapter sends the user name and password over an HTTP connection in plain text.

- **Digest.** The HTTP adapter sends passwords in an encrypted format over the HTTP connection.

- **Kerberos.** Neither the user name nor the password is sent over an HTTP connection. The HTTP adapter uses the credentials of the process under which the HTTP send adapter runs for this authentication type.

Additionally, the HTTP send adapter can provide a client Secure Sockets Layer (SSL) certificate to the Web server if the server requires or accepts it.

Client Certificates for the HTTP Send Adapter

The HTTP send adapter can establish a secure connection with servers that accept or require client certificates. If a client certificate is specified, the HTTP send adapter uses the certificate when connecting with servers that require or accept client certificates. If the client certificate is not specified and the destination server requires client certificates, the HTTP send adapter fails to send the message and follows the standard retry logic.

The HTTP send adapter uses the client certificate from the personal store of the account under which the BizTalk Server process is running. The certificate is specified by its thumbprint. If the HTTP send adapter fails to load the certificate for any reason, the message that it was sending is suspended.

Single Sign-On Support for the HTTP Adapter

You can configure Enterprise Single Sign-On (SSO) for use with the HTTP receive location or send port by using BizTalk Explorer. This topic describes how SSO works with the HTTP adapter.

Single Sign-On Support for the HTTP Receive Location

When an HTTP request is received by Microsoft Internet Information Services (IIS) from a Web client, IIS authenticates the user. The Internet Server Application Programming Interface (ISAPI) extension impersonates the Microsoft Windows user and then calls the SSO credential store to obtain an encrypted ticket. This ticket is stored as the **SSOTicket** property in the context of the message.

In the pass-through scenario, the BizTalk Messaging Engine directs the message to the MessageBox database. When the adapter receives the message from the MessageBox database, the HTTP adapter calls the **ISSOTicket.RedeemTicket Method (COM)** with the encrypted ticket along with the application name to retrieve the back-end credentials from the SSO store. The HTTP adapter then uses the external credentials to connect to the back-end system and process the request. For more information about the affiliate applications, see SSO Affiliate Applications

In the scenario where an orchestration invokes the adapter, the BizTalk Messaging Engine sends this message to the MessageBox database. The orchestration should ensure that both the **SSOTicket** context property and the **Microsoft.BizTalk.XLANGs.BTXEngine.OriginatorSID** context property of the message that contains the ticket are maintained. When the adapter receives this message from the MessageBox database, the adapter calls **RedeemTicket** with the encrypted ticket to retrieve the back-end credentials from the SSO store. The user designing the schedule should specifically copy this property to the message.

Single Sign-On Support for the HTTP Send Adapters

If SSO is enabled, when an HTTP send port receives a message with the **Secure** property, it calls the SSO server to validate and redeem the ticket for an affiliate application. The administration application, affiliate administrators, or SSO administrators for the affiliate application can call SSO to redeem a ticket. SSO then decrypts the ticket and obtains the back-end credentials. The pass-through and orchestration scenario are the same as for the HTTP send port.

By default, SSO is disabled for the HTTP send port. For more information about enabling SSO for the HTTP send port, see Configuring an HTTP Send Port .

To correctly implement Single Sign On support for the HTTP receive and send adapter the following conditions must be met:

- The same user account must be specified in the following places:
  - The application pool identity (IIS 6.0) or hosting COM+ application identity (IIS 5.0) for the IIS virtual directory that is monitored by the HTTP receive adapter. For more information about configuring IIS for HTTP receive locations, see How to Configure IIS for an HTTP Receive Location .
  - The logon credentials used for the isolated host instance that the HTTP adapter is running in. For information about how to configure the logon credentials for a host instance, see How to Modify Host Instance Properties.
- The isolated host that the HTTP adapter is using must be configured as Authentication Trusted. For information about how to configure a host as Authentication Trusted, see How to Modify Host Properties.

Negative Acknowledgment (NACK) Messages Generated for Failed Transmissions by the HTTP or SOAP Adapter

When a message is successfully transmitted, the BizTalk Messaging Engine publishes an associated acknowledgment (ACK) message to the MessageBox if delivery notifications are enabled. Likewise, when a message is suspended by the BizTalk Messaging Engine or an orchestration is suspended by the orchestration engine, BizTalk Server publishes an associated negative acknowledgment (NACK) message to the MessageBox. The NACK message contains context properties and a message body part that consists of a SOAP fault. If the NACK message is generated due to a failed transmission from the HTTP or SOAP adapter, the SOAP fault contains the Headers element and the Body element of the response

from the destination Web server. The following is an example of the SOAP fault in a NACK generated for a failed HTTP transmission:

To subscribe to a NACK message, you can do one of the following:

- Create a send port with a filter for the appropriate message context property. See **Message Context Properties** for a listing of system message context properties including those related to message acknowledgment.

- Send from an orchestration port marked with **Delivery Notification = Transmitted**. If an orchestration port is marked with **Delivery Notification = Transmitted**, the orchestration will wait until it receives either an ACK or a NACK for the message that was transmitted. If a NACK is generated then it will be routed to the orchestration and the orchestration will throw a DeliveryFailureException. The DeliveryFailureException is deserialized from the SOAP fault that is contained within the NACK message body. To retrieve the exception message string from the SOAP fault that is returned to the orchestration, cast the DeliveryFailureException to a SoapException and then access the InnerXml from the SOAP Detail section. The following code sample demonstrates how to do this:

Configuring the HTTP Adapter

This section describes how to configure an HTTP adapter.

In This Section

- How to Configure an HTTP Receive Handler
- How to Configure IIS for an HTTP Receive Location
- How to Configure an HTTP Receive Location
- How to Configure an HTTP Send Handler
- Configuring an HTTP Send Port
- HTTP Adapter Configuration and Tuning Parameters
- HTTP Adapter Property Schema and Properties

How to Configure an HTTP Receive Handler

To configure the general properties for an HTTP receive handler

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **HTTP,** in the right pane, right-click the receive handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.

4. Click **Properties** to access the **Batch size** property for the HTTP receive handler.

5. Enter a value from 1 to 256 and click **OK**.

6. Click **OK**.

If this receive handler is going to be used for two way/request-response receive locations then you can minimize latency by following these steps:

- Set the **Batch size** property to a value of 1.

- Change the **MaxReceiveInterval** value associated with the **Messaging Isolated** entry in the **adm_ServiceClass** table of the BizTalk Management database to a value of 25.

- Restart the IIS Application Pool(s) associated with any HTTP receive functions that you have configured.

The logon account for the **BizTalkServerIsolatedHost** host instance must have Read and Write permissions to the temp directory or directories to dynamically compile the code-behind files used by the HTTP receive function. Use the following procedure to grant permissions.

To grant the account for the BizTalkServerIsolatedHost host instance Read and Write permissions to the temp directory of your BizTalk server

1. Click **Start**, click **Run**, type **CMD**, and press ENTER.

2. At the command prompt, type **set TEMP** and press ENTER to display the directory associated with the **TEMP** environment variable.

3. At the command prompt, type **set TMP** and press ENTER to display the directory associated with the **TMP** environment variable.

Grant the account that is specified as the logon account for the **BizTalkServerIsolatedHost** host instance Read and Write permissions to the directory or directories associated with the **TEMP** and **TMP** environment variables. To determine the logon account for the **BizTalkServerIsolatedHost** instance, in the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, expand **Host Instances**, right-click the **BizTalkServerIsolatedHost** host instance in the right pane, and then click **Properties**. The logon account used for the host instance is listed next to the **Logon** label.

How to Configure IIS for an HTTP Receive Location

Depending on which version of Microsoft Windows you are using, you will have to configure Microsoft Internet Information Services (IIS) differently to work with the HTTP adapter receive location.

If your operating system is Microsoft Windows Server 2003, IIS 6.0 provides two different application isolation modes to protect Web applications. Worker process isolation mode is the default mode, but IIS 5.0 is also provided for backward compatibility. You can configure the HTTP adapter receive location to work with either mode, but worker process isolation mode is recommended for its improved security functionality.

If your operating system is Windows 2000 Server, you must use IIS 5.0 isolation mode, and configure your HTTP receive location accordingly.

To configure the IIS 6.0 worker process isolation mode to work with the HTTP adapter receive location

1. Click **Start**, point to **Settings**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**.
3. In Administrative Tools, double-click **Internet Information Services**.
4. In Internet Information Services (IIS) Manager, right-click **Web Service Extensions**, and then click **Add a new Web service extension**.
5. In the **New Web Service Extension** dialog box, do the following.

Use this	To do this
Extension Name	Type the name of the new Web service extension.
Required files	Click **Add**, and then click **Browse**. Browse to the **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\HttpReceive** directory (or *<drive***>:\Program Files (x86)\Microsoft BizTalk Server 2006\HttpReceive64** directory on 64 bit machines), select **BTSHTTPReceive.dll**, click **Open**, and then click **OK**.
Set extension status to Allowed	Select this check box.

6. Click **OK**.
7. Right-click **Application Pools**, point to **New**, and then click **Application pool**.

8. In the **Add New Application Pool** dialog box, in the **Application pool ID** box, type a name for the application pool, and then click **OK**.

   The new application pool appears in the list of **Application Pools**.

9. Right-click the new application pool, and then click **Properties**.

10. In the **<***Application Pool name***> Properties** dialog box, on the **Identity** tab, do the following.

Use this	To do this
Configurable	Select this property.
User name	Type a user name for the BizTalk Isolated Hosts group and the IIS_WPG group. This user account must have access to the BizTalk Management database, so do not create the application pool to run under the IWAM_<*servername*> user account, which is the default.
Password	Type the password for the user.

11. Click **OK**.

12. Expand the **Web Sites** node, right-click the **Default Web Site** node, point to **New**, and then click **Virtual Directory**.

13. In the Virtual Directory Creation Wizard, on the **Welcome** page, click **Next**.

14. On the **Virtual Directory Alias** page, in the **Alias** box, type the alias to associate with the virtual directory, and then click **Next**.

15. On the **Web Site Content Directory** page, click **Browse**.

16. In the **Browse For Folder** dialog box, navigate to the **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\HttpReceive** directory, click **OK**, and then click **Next**.

17. On the **Access Permissions** page, select **Read** and **Execute**, clear all other check boxes, and then click **Next**.

18. On the **Completion** page, click **Finish** to close and complete the wizard.

    The new virtual directory appears under the list of Default Web Sites in Internet Information Services (IIS) Manager.

19. Right-click the virtual directory, and then click **Properties**.

20. In the **<***Virtual Directory***> Properties** dialog box, on the **Virtual Directory** tab, do the following.

Use this	To do this
Execute Permissions	Ensure that the property is set to **Scripts and Executables**.
Application Pool	Select the new application pool created earlier in this procedure.

21. Click **OK**.

22. On the **File** menu, click **Exit**.

To configure the IIS 5.0 isolation mode to work with the HTTP adapter receive location

1. Click **Start**, and then click **Control Panel**.

2. In Control Panel, double-click **Administrative Tools**.

3. In Administrative Tools, double-click **Internet Information Services**.

4. In Internet Information Services, expand **<***computer name***> (local computer)**, and **Web Sites**.

5. Right-click **Default Web Site**, point to **New**, and then click **Virtual Directory**.

6. In the Virtual Directory Creation Wizard, on the **Welcome** page, click **Next**.

7. On the **Virtual Directory Alias** page, in the **Alias** box, type the alias to associate with the virtual directory, and then click **Next**.

8. On the **Web Site Content Directory** page, click **Browse**.

9. In the **Browse For Folder** dialog box, browse to the **<***drive***>:\Program Files\Microsoft BizTalk Server 2006\HttpReceive** directory, click **OK**, and then click **Next**.

10. On the **Access Permissions** page, select **Read** and **Execute**, clear all other check boxes, and then click **Next**.

11. On the **Completion** page, click **Finish** to close and complete the wizard.

    The new virtual directory appears under the list of Default Web Sites in Internet Information Services.

12. Right-click the virtual directory, and then click **Properties**.

13. In the **<***Virtual Directory***> Properties** dialog box, on the **Virtual Directory** tab, do the following.

Use this	To do this
Execute Permissions	Ensure that the property is set to **Scripts and Executables**.
Application Protection	Change the value of the property to **High (Isolated)** or **Medium (Pooled)**. Medium is not recommended because it imposes a security risk by opening BizTalk databases to all the IIS applications that run with that protection level. Setting this property to High creates a separate COM+ application to host the virtual directory. Setting this property to Medium means that the virtual directory is hosted by the IIS Out-Of-Process Pooled Applications.

14. Click **OK**.

15. On the **File** menu, click **Exit**.

16. Change to the Administrative Tools window, and double-click **Component Services**.

17. In the Component Services window, expand **Component Services**, expand **Computers**, expand **My Computer**, and then expand **COM+ Applications**.

18. If the **Application Protection** property is set to **Medium**, right-click **IIS Out-of-Process Pooled Applications**, and then click **Properties**.

    –Or–

    If the **Application Protection** property is set to **High**, right-click **IIS_{Default Web Site//root/virtual directory name}**, and then click **Properties**.

19. Right-click the virtual directory and then click **Properties**.

20. On the **Identity** tab, configure the account properties for the user that belongs to the BizTalk Isolated Host Users group. This user account must have access to the BizTalk Management database, so do not create the application pool to run under the IWAM_<*server name*> user account, which is the default.

21. Create an HTTP receive location using BizTalk Explorer. Specify the **Virtual directory plus ISAPI extension** property to be equal to **/<virtual directory>/BTSHTTPReceive.dll<query string>**. For instructions about configuring the HTTP receive location

How to Configure an HTTP Receive Location

You can set HTTP receive location adapter variables either programmatically or by using the BizTalk Server Administration console. If properties are not set in the receive location, the default receive handler values set in the BizTalk Server Administration console are used.

How to Configure an HTTP Receive Location Programmatically

The HTTP adapter stores its configuration information in the BizTalk Management database (also known as the Configuration database). The configuration is stored in a custom XML property bag.

The BizTalk Explorer object model exposes the **IReceiveLocation** configuration interface, which has a **TransportTypeData** read/write property. This property accepts the HTTP receive location configuration property bag in a name-value pair XML string.

Setting the **TransportTypeData** property of the **IReceiveLocation** is not required. If it is not set, the default values for the HTTP receive location configuration are used. The following table lists the default values, and also lists the configuration properties that you can set in the BizTalk Explorer object model for the HTTP receive location.

Property name	Type	Description	Restrictions	Comments
ResponseContentType	string	Content type of the HTTP response messages that the HTTP adapter sends back to clients from this receive location. This property is valid only for request-response receive ports and is ignored for one-way receive ports.	String **Minimum length:** 0 **Maximum length:** 256	Default value: Text/XML
LoopBack	Boolean	Specifies that the request message received on this location will be routed either to a send port or back to the receive location to be sent as a response. This property is valid only for request-response receive ports. It is ignored for one-way receive ports.	None	**Default value:** False
ReturnCorrelationHandle	Boolean	Specifies that the correlation token of submitted message that the HTTP adapter sends on HTTP response to the client if the submission is successful. This property is valid only for one-way receive ports and is ignored for request-response receive ports.	None	**Default value:** True
SuspendFailedRequests	Boolean	Specifies whether to suspend failed HTTP requests. A value of True indicates to suspend the failed request and send an	None	**Default value:** False

		"Accepted" status code (202) to the client for one-way receive ports or an "Error" status code (500) to the client for two-way receive ports.		
UseSSO	Boolean	Specifies whether the HTTP adapter will issue the SSO ticket to messages that arrive on this receive location.	None	**Default value:** False

The format of the XML string to set these properties is as follows:

How to Configure an HTTP Receive Location with the BizTalk Server Administration Console

To configure the receive location by using the BizTalk Server Administration console, use the following procedure.

To configure variables for an HTTP receive location

1. Configure Internet Information Services (IIS) to work with HTTP receive locations. For instructions about configuring IIS, see How to Configure IIS for an HTTP Receive Location .

2. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application in which you want to create a receive location.

3. In the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

4. In the **Receive Port Properties** dialog box, in the left pane, select **Receive Locations**, and in the right pane, double-click an existing receive location or click **New** to create a new receive location.

5. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **HTTP** from the drop-down list and then click **Configure**.

6. In the **HTTP Transport Properties** dialog box, do the following:

Use this	To do this
Virtual directory plus ISAPI extension	Specify the name of the virtual directory where you post the messages received by the HTTP/HTTPS receive location. The virtual directory includes the name of the receive location DLL and an optional query string. Examples of virtual directory names are:

	/<virtual directory>/BTSHTTPReceive.dll /<virtual directory>/BTSHTTPReceive.dll?Purchase%20Order This location must not contain more than one BTSHTTPReceive.dll ISAPI extension, including all subfolders. **Type:** String **Maximum length:** 256
Public Address	Specify the fully qualified URI for this receive location. The value for this property is a combination of the server name and the virtual directory. The BizTalk Messaging Engine exposes this address to external partners. The specified URI should designate the public Web site URL for trading partners to connect to when sending messages to BizTalk Server. This information is optional and is not used by BizTalk Server. This parameter is available to allow administrators to document the public URL that the receive location is tied to. **Type:** String **Minimum length:** 0 **Maximum length:** 256
Return Content type	Specify the content type of HTTP response messages that the receive location sends back to clients. This property is valid only for request-response receive locations. **Default value:** text/xml **Type:** String **Minimum length:** 0 **Maximum length:** 256
Loopback	Define that the request message received on this location is routed either to a send port or back to this receive location to be sent as a response. This property is valid only for request-response receive locations. **Default value:** False **Type:** Boolean

Return correlation handle on success (One way port only)	Define that if successful, the receive location sends the correlation token of the submitted message on the HTTP response to the client. This property is valid only for one-way receive locations. **Default value:** True **Type:** Boolean
Use Single Sign On	Indicate that Enterprise Single Sign-On is used. **Default value:** False **Type:** Boolean
Suspend Failed Requests	Indicate whether or not to suspend HTTP requests that fail inbound processing. A value of False indicates to discard the failed request and send an error status code (401 or 500) to the client. A value of True indicates to suspend the failed request and send an "Accepted" status code (200) to the client for one-way receive ports or an "Error" status code (500) to the client for two-way receive ports. **Default value:** False **Type:** Boolean

7. Click **OK** to save settings.

8. Enter the appropriate values in the **Receive Location Properties** dialog box to complete the configuration of the receive location and click **OK** to save settings. For information about the **Receive Locations Properties** dialog box, see How to Create a Receive Location.

How to Configure an HTTP Send Handler

To change global variables for an HTTP send handler

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **HTTP**, in the right pane right-click the send handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the send handler will be associated.

4. On the **General** tab, do the following.

Use this	To do this
Request timeout (sec)	Specify the time-out in seconds when waiting for a response from the server. If set to zero (0), the HTTP adapter calculates the time-out based on the request message size. If you do not provide a value, the value for the handler is used. **Default value:** 0 **Type:** Long **Minimum value:** 0 **Maximum value:** MAX_LONG
Maximum redirects	Specify the maximum redirects allowed for the message being sent. **Default value:** 5 **Type:** Int **Minimum value:** 0 **Maximum value:** 10
Content type	Specify the content type of the request messages. If you do not provide a value, the value for the handler is used. **Default value:** Text/XML **Type:** String **Minimum length:** 0 **Maximum length:** 256

5. On the **Proxy** tab, do the following.

Use this	To do this
Use proxy	Specify whether the HTTP send handler uses the proxy server. **Default value:** False **Type:** Boolean
Server	Specify the proxy server address for this send port. This property requires a value if **Use proxy** is **True**. **Type:** String **Minimum length:** 0 **Maximum length:** 256
Port	Specify the proxy server port for this send port. This property requires a value if **Use proxy** is **True**. **Default Value:** 80 **Type:** Long **Minimum value:** 0 **Maximum value:** 65535
User name	Specify the user name to use for authentication with the proxy server. This property requires a value if **Use proxy** is **True**. **Type:** String **Minimum length:** 0 **Maximum length:** 256
Password	Specify the user password for authentication with the proxy server. This property requires a value if **Use proxy** is **True**. **Type:** String **Minimum length:** 0

	Maximum length: 256

6. Click **OK**.

Configuring an HTTP Send Port

This section describes how to configure an HTTP send port.

In This Section

- How to Configure an HTTP Send Port
- How to Configure an HTTP Send Port With a Remote BizTalk Management Database
- Restrictions on the Destination URL Property

How to Configure an HTTP Send Port

How to Configure an HTTP Send Port Programmatically

The HTTP adapter stores its configuration information in the BizTalk Management database (also known as the Configuration database). You store configuration information in a custom XML property bag. During initialization of the HTTP adapter and during its run time, the server passes the configuration to the adapter as follows:

- For the HTTP send handler, configuration information passes to the adapter by calling the **Load** method of the **IPersistPropertyBag** interface.
- For the HTTP send ports, configuration information passes to the adapter as a set of properties on a message context. The HTTP namespace groups these properties together.

The BizTalk Explorer object model exposes the **ItransportInfo** adapter configuration interface for send ports, which contains the **TransportTypeData** read/write property. This property accepts the HTTP send port configuration property bag as a name/value pair XML string. Note that to set this property in the BizTalk Explorer object model, it must first be set on the **Address** property of the **ITransportInfo** interface.

Setting the **TransportTypeData** property of the **ITransportInfo** interface is not required. If it is not set, the HTTP adapter will use the default values for the HTTP send handler.

If send port configuration properties that duplicate the configuration for the handler are not defined, configuration properties for the handler are used. If the HTTP send handler does not have configuration values, the HTTP send adapter logs an error in the event log and moves the message to the backup adapter.

You can set configuration properties programmatically on a message context. You can set these properties in a BizTalk Server orchestration schedule or in custom pipeline components. The following rules apply when using these properties:

- If the configuration property is set on an orchestration or in a custom pipeline component in a receive pipeline, then:
  - If a message is sent to a static send port, the property value will be overwritten with the value configured for that send port.
  - If a message is sent to a dynamic send port, the property value will not be overwritten.
- If the configuration property is set in a custom pipeline component in a send pipeline, then:
  - The value will not be overwritten regardless of whether the message is sent to a static or dynamic send port.

The following table lists the configuration properties that you can set in the BizTalk Explorer object model for the HTTP send location.

Property name	Type	Description	Restrictions	Comments
RequestTimeout	xs:int	Time-out period of waiting for a response from the server. If set to zero (0), the system calculates the time-out based on the request message size.	**Minimum value:** 0 **Maximum value:** MAX_LONG	**Default value:** 0
ContentType	xs:string	Content type of the request messages	**Minimum length:** 0 **Maximum length:** 256	**Default value:** Text/XML
MaxRedirects	xs:int	Maximum number of times that the HTTP adapter can redirect the request.	**Minimum value:** 0 **Maximum value:** 10	**Default value:** 5

UseHandlerProxySettings	xs:boolean	Specifies whether the HTTP send port will use the proxy configuration for the send handler.	None	**Default value:** True
UseProxy	xs:boolean	Specifies whether the HTTP adapter will use the proxy server.	None	**Default value:** False This property is ignored if **UseHandlerProxySettings** is **True**.
ProxyName	xs:string	Specifies the proxy server name.	**Minimum length:** 0 **Maximum length:** 256	**Default value:** Empty The HTTP send adapter ignores this property if the **UseHandlerProxySettings** property is set to **True**. Otherwise, the HTTP send adapter uses this property only if **UseProxy** is **True**. This property is required if **UseProxy** is **True**.
ProxyPort	xs:int	Specifies the proxy server port.	**Minimum value:** 0 **Maximum value:** 65535	**Default value:** 80 The HTTP send adapter ignores this property if **UseHandlerProxySettings** is **True**. Otherwise, HTTP send adapter uses this property only if **UseProxy** is **True**. This property is required if **UseProxy** is **True**.
ProxyUsername	xs:string	Specifies the user name for authentication with the proxy server.	**Minimum length:** 0 **Maximum length:** 256	**Default value:** empty The HTTP send adapter ignores this property if **UseHandlerProxySettings** is **True**. Otherwise, HTTP send adapter uses this property only if **UseProxy** is **True**.

ProxyPassword	xs:string	Specifies the user password for authentication with the proxy server.	**Minimum length:** 0 **Maximum length:** 256	**Default value:** empty The HTTP send adapter ignores this property if **UseHandlerProxySettings** is **True**. Otherwise, HTTP send adapter uses this property only if **UseProxy** is **True**.
AuthenticationScheme	xs:string	Type of authentication to use with the destination server.	None	**Valid values:** • **Anonymous (Default)** • **Basic** • **Digest** • **Kerberos**
Username	xs:string	User name to use for authentication with the server.	**Minimum length:** 0 **Maximum length:** 256	**Default value:** Empty This value is required if you select **Basic** or **Digest** authentication. The HTTP adapter ignores the value of this property if **UseSSO** is **True**.
Password	xs:string	User password to use for authentication with the server.	**Minimum length:** 0 **Maximum length:** 256	**Default value:** empty This value is required if you select **Basic** or **Digest** authentication. The value of this property is ignored if **UseSSO** is **True**.
EnableChunkedEncoding	xs:boolean	Specifies whether or not chunked encoding is used by the HTTP adapter	None	**Default value:** True
Certificate	xs:string	Thumbprint of the client SSL certificate.	**Minimum length:** 0 **Maximum**	**Default value:** Empty

			length: 59	
UseSSO	xs:boolean	Specifies if SSO will be used for the send port.	None	**Default value:** False
AffiliateApplicationName	xs:string	Name of the affiliate application to use for SSO.	**Minimum length:** 0 **Maximum length:** 256	**Default value:** empty Required if **UseSSO** is **True**.

The following code shows the XML string to use to set these properties:

How to Configure an HTTP Send Port with the BizTalk Server Administration Console

You can set HTTP send port adapter variables in the BizTalk Server Administration console. If properties are not set for the send port, the default send handler values set in the BizTalk Server Administration console are used.

To configure variables for an HTTP send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. See How to Create a Send Port for more information. Configure all of the send port options and specify **HTTP** for the **Type** option in the **Transport** section on the **General** tab.

2. On the **General** tab, in the **Transport** section, click the **Configure** button next to **Type**.

3. In the **HTTP Transport Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Destination URL	Required. Specify the address to send HTTP requests. Include query strings appended to the base URL. **Type:** String **Maximum length:** 256 For more information, see Restrictions on the Destination URL Property .
Enable chunked encoding	Specify to use chunked encoding. If this option is enabled, the HTTP adapter will use HTTP chunked encoding with maximum chunk size of 8 KB. Chunked encoding is implicitly disabled if the HTTP send handler is configured to **Use**

	proxy. **Type:** Boolean **Default Value:** True
Request timeout (sec)	Specify the time-out in seconds for the HTTP/HTTPS transmission. If the HTTP adapter does not receive the response within this time, the service logs the error and resubmits the message based on the retry infrastructure. If set to zero (0), the BizTalk Messaging Engine calculates the time-out based on the request message size. If you do not provide a value, the value for the handler is used. **Type:** Long **Minimum value:** 0 **Maximum value:** MAX_LONG
Maximum redirects	Specify the maximum redirects allowed for the message being sent. **Default value:** 5 **Type:** Int **Minimum value:** 0 **Maximum value:** 10
Content type	Specify the content type of the request messages. If this value is not set, the value for the handler is used. **Type:** String **Minimum length:** 0 **Maximum length:** 256

4. In the **HTTP Transport Properties** dialog box, on the **Proxy (Handler override)** tab, do the following:

Use this	To do this
Use Handler's default proxy configuration	Specify that the send port configuration must use the proxy settings specified for the HTTP send handler. This is the default setting.
Do not use proxy	Specify whether the HTTP send handler uses the proxy server. If selected, the HTTP send handler for this send port does not use the proxy server.
Use proxy	Specify whether the HTTP send handler uses the proxy server. If selected, the HTTP send handler uses the proxy server.
Server	Specify the proxy server address for this send port. This property only requires a value if **Use proxy** is selected. **Type:** String **Minimum length:** 0 **Maximum length:** 256
Port	Specify the proxy server port for this send port. This property only requires a value if **Use proxy** is selected. **Default Value:** 80 **Type:** Long **Minimum value:** 0 **Maximum value:** 65535
User name	Specify the user name for authentication with the proxy server. This property only requires a value if **Use proxy** is selected. **Type:** String **Minimum length:** 0 **Maximum length:** 256

Password	Specify the user password for authentication with the proxy server. This property only requires a value if **Use proxy** is selected. **Type:** String **Minimum length:** 0 **Maximum length:** 256

5. In the **HTTP Transport Properties** dialog box, on the **Authentication** tab, do the following:

Use this	To do this
Authentication Type	Specify the type of authentication to use with the destination server. Valid options are: • **Anonymous** • **Basic** • **Digest** • **Kerberos** **Default Value:** Anonymous
Credentials	Specify the type of credentials to use. Only available if the **Authentication Type** is **Basic** or **Digest**. Valid options are: • **Do Not Use Single Sign-On** **User name:** The user name to use for authentication with the destination server. If the **Authentication Type** property is **Anonymous** or **Kerberos**, this option is disabled. This property requires a value if **Basic** or **Digest** is selected, and Enterprise Single Sign-On is not used. **Minimum length:** 0 **Maximum length:** 256

	Password: The password to use for authentication with the destination server. If the **Authentication Type** property is **Anonymous** or **Kerberos**, this option is disabled. This property requires a value if **Basic** or **Digest** is selected, and Single Sign-On is not used. **Minimum length:** 0 **Maximum length:** 256 • **Use Single Sign-On** Specify whether to use Single Sign-On to retrieve client credentials for authentication with the destination server. **Affiliate Application** Specifies the affiliate application to use for Single Sign-On. Choose the applications that you want to include in Single Sign-On. **Minimum length:** 0 **Maximum length:** 256
SSL client certificate thumbprint	Specify the thumbprint of the client certificate to use for establishing a Secure Sockets Layer (SSL) connection. **Minimum length:** 0 **Maximum length:** 59

6. Click **OK** and **OK** again to save settings.

How to Configure an HTTP Send Port With a Remote BizTalk Management Database

The HTTP send port has port-level configuration that may require a user name and password. When you export this configuration to a binding file, the system does not keep the password in the binding file for security reasons. As a result, in cases when user names and passwords are configured in the source environment, after you import the binding file into the destination environment (with a different BizTalk Management database), you must use BizTalk Explorer (or the BizTalk Explorer object model) to specify passwords for these port configurations. If the destination environment does not have Microsoft Visual Studio 2005 installed, use the following steps to complete the HTTP send port configuration for the

destination environment in BizTalk Explorer. Perform the following steps on the computer that is connected to the destination environment.

To configure the HTTP send port with a remote BizTalk Management database

1. Click **Start**, point to **Programs**, point to **Microsoft Visual Studio 2005**, and then click **Microsoft Visual Studio 2005**.
2. On the **View** menu, click **BizTalk Explorer**.
3. In BizTalk Explorer, right-click the **BizTalk Configuration Databases** node, and then click **Add Database**.
4. In the **Add Database** dialog box, specify the management database of the production environment. A new node appears under the root node.
5. Expand the newly added node, select the send port that needs reconfiguration, and specify a password for the send port.

Restrictions on the Destination URL Property

The destination URL is a string that specifies the address of the HTTP server where you want to send messages using the HTTP protocol.

The following rules and restrictions apply to the destination URL property:

- You must always specify the destination URL property in the following format:

  http[s]://<host>[:<port>][/<path>[/<file>[?<query-string>]]]

- The whole string may or may not be URI encoded.
- The whole string, except the query string, cannot contain any of the following characters: < > : \ | " ? *.
- The property is not case-sensitive.
- The length of the string must not exceed 256 characters.

HTTP Adapter Configuration and Tuning Parameters

Several configuration and tuning parameters are accessible for the HTTP adapter through registry key entries and through the modification of the BTSNTSvc.exe.config file that is located in the root BizTalk Server installation directory.

Registry Settings That Affect HTTP Adapter Performance

The following table describes the registry settings that affect the performance of the HTTP adapter. Note that by default there are no HTTP adapter keys in the registry, so the HTTP

adapter uses the default settings. If it is necessary to change the default settings, you need to create the following registry keys under the following locations in the registry:

- DisableChunkEncoding, RequestQueueSize, and HttpReceiveThreadsPerCpu must be defined in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BTSSvc.3.0\HttpReceive.

- HttpOutTimeoutInterval, HttpOutInflightSize, and HttpOutCompleteSize must be defined in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BTSSvc{GUID} where GUID is the ID of the host for the HTTP send handler.

Key name	Type	Default	Explanation
DisableChunkEncoding	DWORD	0	Regulates whether or not the HTTP receive adapter uses chunked encoding when sending responses back to the client. Set to a nonzero value to turn off chunked encoding for HTTP receive adapter responses. **Minimum value:** 0 **Maximum value:** Any nonzero value
RequestQueueSize	DWORD	256	Defines the number of concurrent requests that the HTTP receive adapter processes at one time. **Minimum value:** 10 **Maximum value:** 2048
HttpReceiveThreadsPerCpu	DWORD	2	Defines the number of threads per CPU that are allocated to the HTTP receive adapter. **Minimum value:** 1 **Maximum value:** 10
HttpOutTimeoutInterval	DWORD	2000	Defines the interval in seconds that the HTTP send adapter will wait before timing out. **Minimum value:** 500 **Maximum value:** 10000000
HttpOutInflightSize	DWORD	100	This is the maximum number of concurrent HTTP requests that BizTalk Server HTTP send adapter instance will handle.

			The recommended value for latency is between 3 to 5 times that of the **maxconnection** configuration file entry discussed below. **Minimum value:** 1 **Maximum value:** 1024
HttpOutCompleteSize	DWORD	5	**Minimum value:** 1 **Maximum value:** 1024

Configuration File Entry to Govern the Number of Concurrent Connections Made by the HTTP Send Adapter to a Particular Destination Server

The number of concurrent connections that the HTTP adapter opens for a particular destination server can be configured by making an entry in the BTSNTSvc.exe.config file that is located in the root BizTalk Server installation directory.

HTTP Adapter Property Schema and Properties

Namespace: http://schemas.microsoft.com/BizTalk/2003/http-properties

Name	Type	Description
ProxyName	xs:string	Specifies the proxy server name.
ProxyPort	xs:int	Specifies the proxy server port.
UseHandlerProxySettings	xs:boolean	Specifies whether the HTTP send port uses the proxy configuration for the handler.
UseProxy	xs:boolean	Specifies whether HTTP adapter uses the proxy server.
RequestTimeout	xs:int	Time-out period of waiting for a response from the server. If this property is set to zero (0), the system calculates the time-out on the request message size.
Username	xs:string	The user name to use for authentication with the server.
Password	xs:string	The user password to use for authentication with the server.
ProxyUsername	xs:string	Specifies the user name for authentication with the proxy server.

ProxyPassword	xs:string	Specifies the user password for authentication with the proxy server.
MaxRedirects	xs:int	The maximum number of times that the HTTP adapter will redirect the request.
ContentType	xs:string	Content type of the request messages.
AuthenticationScheme	xs:string	Type of authentication to use with the destination server.
Certificate	xs:string	Thumbprint of client SSL certificate.
UseSSO	xs:boolean	Specifies whether the HTTP send port will use SSO.
AffiliateApplicationName	xs:string	Name of affiliate application to use for SSO.
InboundHttpHeaders	xs:string	Contains the HTTP headers from the inbound HTTP request.
SubmissionHandle	xs:string	Contains the BizTalk Server correlation token (GUID) for the request message.
EnableChunkedEncoding	xs:boolean	Specifies whether or not chunked encoding is used by the HTTP adapter.
UserHttpHeaders	xs:string	Contains the customized headers contained in the HTTP request or response message The value of the **UserHttpHeaders** property must have the following format: You can modify the following five standard HTTP headers by using the **UserHttpHeaders** property: • Accept • Referrer • Expect • If-Modified-Since • User-Agent

HTTP Adapter Security Recommendations

You use the HTTP adapter to exchange information between BizTalk Server and an application by means of the Hypertext Transfer Protocol (HTTP). Applications can send messages to a server by sending HTTP POST or HTTP GET requests to a specified HTTP URL. For more information about the HTTP adapter, see HTTP Adapter. It is recommended that you use the following guidelines for securing and deploying the HTTP adapter in your environment:

- Ensure you configure the Internet Information Services (IIS) settings for the HTTP adapter. For more information, see Configuring Internet Information Services for HTTP Receive Locations.

- If you use IIS 6.0, ensure you follow the IIS 6.0 recommendations for configuring application isolation. For more information, see the Microsoft TechNet Web site at http://go.microsoft.com/fwlink/?LinkId=25222.

- If you use IIS 5.0 or 5.1, ensure you follow the IIS 5.0 recommendations for securing IIS 5.0. For more information, see the Microsoft TechNet Web site at http://go.microsoft.com/fwlink/?LinkId=24776.

- When you use Basic Authentication, or when you do not use encryption at the message level, it is recommended to use Secure Sockets Layer (SSL) for both receiving and sending messages to ensure that an unauthorized person cannot sniff the user credentials.

- It is recommended to use Windows integrated authentication for both sending and receiving messages.

- It is recommended that you do not rename, copy, or move the ISAPI extension file. This ensures that the security update installers can correctly apply any potential security updates pertinent to this file.

- You should use strong discretionary access control lists (DACLs) for the directory containing the ISAPI extension file and for the virtual directory that you create for receiving messages. Members of the BizTalk Isolated Host group for the host running the HTTP adapter need read and execute permissions, and the users that the HTTP adapter authenticates need read permissions on these directories.

- When you use SSL client certificates with the HTTP send adapter, you must manually configure these certificates. For more information about configuring the SSL client certificates, see **Configuring an HTTP Send Port By Using BizTalk Explorer**.

- Just like other BizTalk Server components, it is recommended you do not put the HTTP adapter in the perimeter network. If you do, you have to open ports from the perimeter network to the data domain for SQL Server traffic to the MessageBox database, which is risk-prone. It is recommended you configure the HTTP adapter in the processing domain (that is, not the perimeter network). You can then configure the outmost firewall (FW4) to forward HTTP requests through the firewall in the processing domain (FW3). In this case, you do not need IIS in the perimeter network. This mechanism is called reverse proxy. (The ISA implementation is called Web Publishing.)

- When you create an application pool for an HTTP receive location, you must configure it to run under an account that is a member of the Windows group for the isolated host running the HTTP receive adapter and the Internet Information Services Worker Process group (IIS_WPG group). You must then use the BizTalk Server Administration console to configure the host instance for the HTTP receive adapter to use this account. If you change the account for the IIS_WPG group, you must ensure you also update the host instance to run under the new account. For more information, see Configuring Internet Information Services for HTTP Receive Locations.

MQSeries Adapter

The MQSeries adapter serves as a bridge between Microsoft BizTalk Server and IBM MQSeries servers, enabling you to use a full range of options in creating your business processes.

In This Section

- What Is the MQSeries Adapter?
- Configuring the MQSeries Adapter
- Walkthrough: Creating a BizTalk Application That Uses the MQSeries Adapter
- MQSeries Adapter Batching and Transaction Handling
- Correlating Messages Using Request-Reply
- MQSeries Adapter Custom Headers
- Analyzing MQSeries Adapter Errors with the Trace Tools
- Ordered Delivery of Messages with the MQSeries Adapter

What Is the MQSeries Adapter?

With the MQSeries adapter you can send and receive messages to MQSeries systems using Microsoft BizTalk Server 2006.

The adapter relies on MQSeries Server for Windows. This design guarantees reliable messaging because MQSeries Server for Windows supports Microsoft Distributed Transaction Coordinator (MSDTC).

The adapter supports clustered MQSeries Servers and clustered MQSeries Queue Managers, and also clustered BizTalk servers.

You can do the following with the MQSeries adapter:

- Send messages to MQSeries remote definition queues, local queues, transmission queues, and alias queues from BizTalk Server.

- Receive messages from MQSeries transmission queues, local queues, and alias queues.
- Send and receive messages from MQSeries Server for Windows (MQSeries Server can run on the same computer as BizTalk Server or on a remote installation). You only have to deploy one copy of MQSAgent (the COM+ component of the adapter) to support all your BizTalk Server installations.
- Poll MQSeries Server with a wait interval.
- Use dynamic send ports to control the adapter.
- Dynamically create queues at run time.
- Dynamically receive messages from queues based upon MQSeries MatchOptions
- Map context properties to header properties for both transmitting and receiving messages. You can get and set MQSeries header properties (including MQMD, MQXQH, MQCIH, and MQIIH) through BizTalk Server context properties.
- Enable correlation with either BizTalk Server 2006 or MQSeries Server creating the correlation identifier.
- Request transactional and nontransactional delivery of messages for send and receive.

In This Section

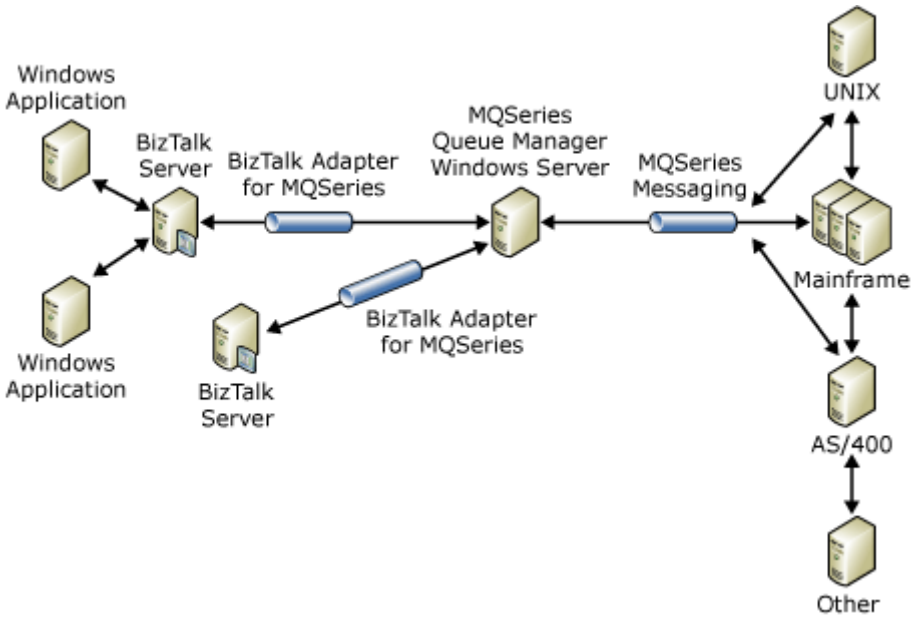
- Components of the MQSeries Adapter
- MQSeries Adapter Architecture
- Using the MQSeries Adapter
- MQSeries Adapter Security

Components of the MQSeries Adapter

The MQSeries adapter uses two components to facilitate document transfer between BizTalk Server and MQSeries Server for Windows.

- **BizTalk component.** Install this component on the same computer as Microsoft BizTalk Server. This component communicates with BizTalk Server.
- **MQSeries component.** Install this component on the MQSeries Server for Windows. MQSeries Server for Windows runs on Microsoft Windows Server 2003, Windows XP Professional, or Windows 2000 Server. This component (referred to as MQSAgent) communicates with IBM MQSeries Server.

The following figure outlines a typical use of the adapter.



The MQSeries adapter is a connectivity solution that lets you use BizTalk Server in an enterprise with MQSeries as the chosen messaging standard. Developing this solution was motivated, in part, by the following issues:

- Accommodating customer requests for simple installation and configuration, and an MQSeries connectivity solution
- Supporting message sizes up to 100 MB
- Providing MQSeries support
- Providing a Plug and Play connectivity solution for MQSeries messages to BizTalk Server
- Bridging the messaging protocol of yesterday and the integration server of today

The MQSeries adapter is a key addition to the BizTalk Server suite of receive services that provide a set of listeners for various communication protocol standards. The listeners attach a protocol, for example HTTP, FTP, or MQSeries, to an enterprise application integration (EAI), business-to-business, or application-to-application integration trading relationship.

MQSeries Adapter Architecture

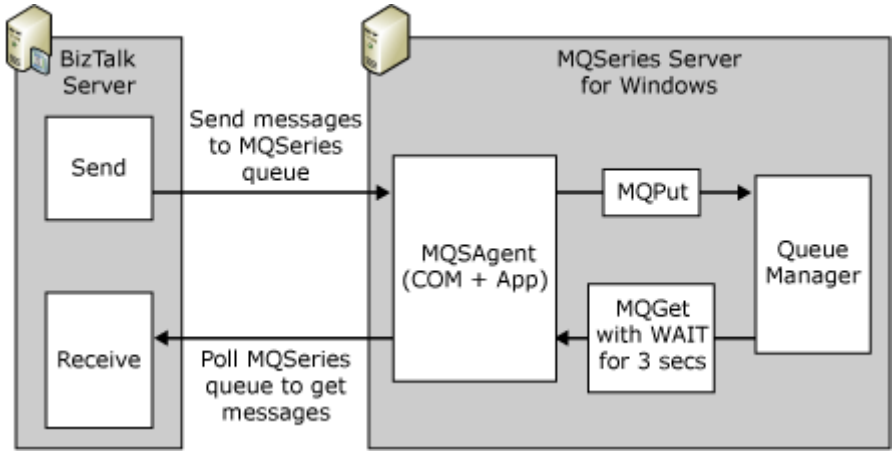
This section introduces the parts of the MQSeries adapter and the ways in which they interact. It also briefly describes some elements of MQSeries queues.

In This Section

- Structure of the MQSeries Adapter
- MQSeries Queues

Structure of the MQSeries Adapter

The MQSeries adapter has two parts: the adapter running under BizTalk Server 2006 and a COM+ application, MQSAgent, running under MQSeries Server for Windows. The following figure shows this relationship.



The adapter communicates with the MQSAgent application. The MQSAgent application, in turn, communicates with MQSeries Server for Windows. You can install the agent on the same computer as the adapter if you install MQSeries Server for Windows on the computer.

The send part of the adapter sends the message to the MQSAgent. MQSAgent then, using **MQPut**, sends the message to the MQSeries Queue Manager.

The receive part of the adapter polls the MQSAgent to see if there are messages. When there is a message, the MQSAgent performs an **MQGet** to retrieve the message. MQSAgent includes a hard-coded three-second wait for retrieving the message from the Queue Manager.

Both the send and receive message actions may occur in transactions. This enables the adapter to roll back the message and, possibly, to retry the send or receive operations. For more information about transactions,.

Because the adapter works across more than one computer, there is a possible security problem. A hostile program could impersonate the agent and capture data.

MQSeries Queues

To send messages to MQSeries queues, you can specify a remote definition queue, an alias queue, a transmission queue, or a local queue in the Queue Manager.

For receiving messages from MQSeries Server, you can specify a transmission queue, an alias queue, or a local queue in the Queue Manager.

Typically, you use a remote queue definition for a send queue, and a transmission queue for receiving messages.

For information about the different MQSeries queues, see the IBM WebSphere MQ documentation.

Using the MQSeries Adapter

The MQSeries adapter serves organizations requiring integration with BizTalk Server 2006 and using MQSeries Server as the primary messaging system.

A network administrator deploys and configures the adapter. A software developer can extend the functionality by creating adapter components.

In This Section

- MQSeries Adapter Deployment Options
- MQSeries Adapter Message Flow
- Using MQSeries Adapter with an Earlier Version of the Adapter
- MQSeries Adapter High Availability

MQSeries Adapter Deployment Options

The MQSeries adapter gives you great flexibility in configuring your hardware. There are at least three main patterns of use:

- BizTalk Server, the adapter, and MQSeries Server for Windows on the same computer.
- BizTalk Server and the adapter on one computer, and MQSeries Server for Windows (including the MQSAgent) on a second computer, which connects to one or more additional computers that are running MQSeries Server.
- Multiple BizTalk Server installations in a group and the adapter, and MQSeries Server (including MQSAgent) on a separate computer.
- The BizTalk Server 2006 version of the MQSeries adapter and the BizTalk Server 2004 version of the MQSeries adapter can work with the same remote MQSeries Server on Windows. This is possible because the MQSAgent (MQSAgent2) COM+ application used with BizTalk Server 2006 can coexist with the MQSAgent COM+ application used with BizTalk Server 2004.
- The adapter functions correctly if you cluster MQSeries Server and the MQSeries Queue Managers.

MQSeries Adapter Message Flow

A message originating from a BizTalk Server 2006 computer is first passed to an MQSeries Server running on Windows. MQSeries Server running on Windows can be on the same computer as the one that runs BizTalk Server. The message is routed through the MQSeries Server for Windows computer to an MQSeries Server host on an operating system such as UNIX. An application then retrieves the message from the MQSeries queue.

A message originating from an application first goes to an MQSeries queue on MQSeries Server. The MQSeries Server forwards the message to the MQSeries Server for Windows computer. BizTalk Server receives the message from the MQSeries Server for Windows computer and forwards it to the appropriate application.

The MQSeries adapter supports the following messaging scenarios.

Scenario	Description
Receive	The adapter receives a message from MQSeries Server, which is passed to BizTalk Server 2006.
Send (Static One-Way Port)	The adapter routes a message that originates from BizTalk Server 2006.
Dynamic Send	Enables the application to select a destination address (URI) at run time.
Dynamic Receive	Enables the application to select a source address (URI) at run time by setting the **MQSeries.DynamicReceive** context property to **Yes** and specifying the dynamic receive address.
Correlation	Messages from the adapter are correlated with specific instances of an orchestration that can handle more than one type of message. MQSeries Server can create the correlation identifier by using solicit-response, or BizTalk Server can create the correlation identifier.

Using MQSeries Adapter with an Earlier Version of the Adapter

The BizTalk Server 2006, BizTalk Server 2006, and BizTalk Server 2002 versions of the MQSeries adapter can all work with the same remote WebSphere MQ Server on Windows. This is possible because the following versions of the COM+ applications used with the MQSeries adapter can coexist on the same WebSphere MQSeries computer:

- **MQSAgent (MQSAgent2) COM+ application** used with the MQSeries Adapter for BizTalk Server 2006.
- **MQSAgent COM+ application** used with the MQSeries Adapter for BizTalk Server 2006.
- **MQHelper COM+ application** used with the MQSeries Adapter for BizTalk Server 2002.

MQSeries Adapter High Availability

You can improve availability in several ways when you use the adapter:

- Set up a fault-tolerant hosting environment for BizTalk Server 2006.
- Use clustered queue managers in IBM WebSphere MQ, server components for Windows platforms.
- Use Windows Clustering with IBM WebSphere MQ.

For information about fault tolerance and BizTalk Server 2006, see **Sample BizTalk Server Architectures** and Planning Your Platform for Fault Tolerance in BizTalk Server 2006 Help.

You can easily create clustered queue managers through the IBM WebSphere MQ snap-in. For information about creating clusters, see the IBM WebSphere MQ documentation. When you use clustered queue managers you refer to them by the name of the cluster. For example, if you used clustered queue managers you would enter the name of the cluster in the Queue Manager box of the Queue Definition dialog box. You would not use the name of one of the individual queue managers. Also, to use clustered queue managers, you must create a queue with the same name under each queue manager. Use this name in the Queue box of the Queue Definition dialog box.

For more information about clustering IBM WebSphere MQ, see the IBM WebSphere MQ documentation. For more information about configuring MSDTC in a Windows cluster environment, see Microsoft Knowledge Base article number 290624, "How to configure MSDTC in a Windows 2000 cluster environment," available at http://go.microsoft.com/fwlink/?LinkId=57579.

There is no requirement to cluster the MQSAgent (MQSAgent2) COM+ application that is used with the BizTalk Server 2006 MQSeries adapter. To provide high availability for this component, install the component on each cluster node. If the COM+ application stops, the next call from the client will start it.

MQSeries Adapter Security

MQSeries adapter security begins with securing your BizTalk and MQSeries servers. For information about securing BizTalk Server, see **Security and Protection** in Microsoft BizTalk Server 2006 Help. For information about MQSeries Server security, see the IBM MQSeries Server documentation.

Using the adapter itself securely requires attention to four areas:

- Choosing the application identity and members for MQSAgent
- Controlling the BizTalk Server accounts using the adapter
- Securing the queue creation scripts
- Making appropriate use of the **SSO Affiliate Application** property

The account assigned to the application identity during configuration should not be an administrator account. Rather, the account should have the minimum required privileges—read and write access to the MQSeries queues.

Make sure that you assign only BizTalk Server accounts using the adapter to the MQSAgent role.

When using exported scripts created during the queue definition process, keep the scripts in a secure area. Only administrators using the scripts should have access.

If your application uses MQCIH and MQIIH header properties to put user credentials in outbound messages, use the **SSO Affiliate Application** property on the **Transport Properties** page

Configuring the MQSeries Adapter

This section describes how to configure the MQSeries adapter.

In This Section

- How to Configure MQSeries Adapter Receive Locations and Send Ports
- How to Configure MQSeries Adapter Send and Receive Handlers
- Using the MQSAgent COM+ Configuration Wizard
- Silent Configuration of the MQSeries Adapter
- MQSeries Adapter Properties

How to Configure MQSeries Adapter Receive Locations and Send Ports

You can configure the MQSeries adapter for both receive locations and send ports.

To configure receive locations and send ports

To create the receive port and receive location:

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application in which you want to create a receive location.

2. Right-click the **Receive Ports** node, click **New,** and point to **One-Way Receive Port**.

3. Enter the appropriate values in the **Port Properties** dialog box. For information about the **Port Properties** dialog box, In the BizTalk Server Administration console, right-click the **Receive Port** node you created and then click **Properties**.

4. In the **Receive Port Properties** dialog box, in the left pane, select **Receive Locations**, and then click **New** in the right pane.

5. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **MQSeries** from the drop-down list, and then click **Configure**.

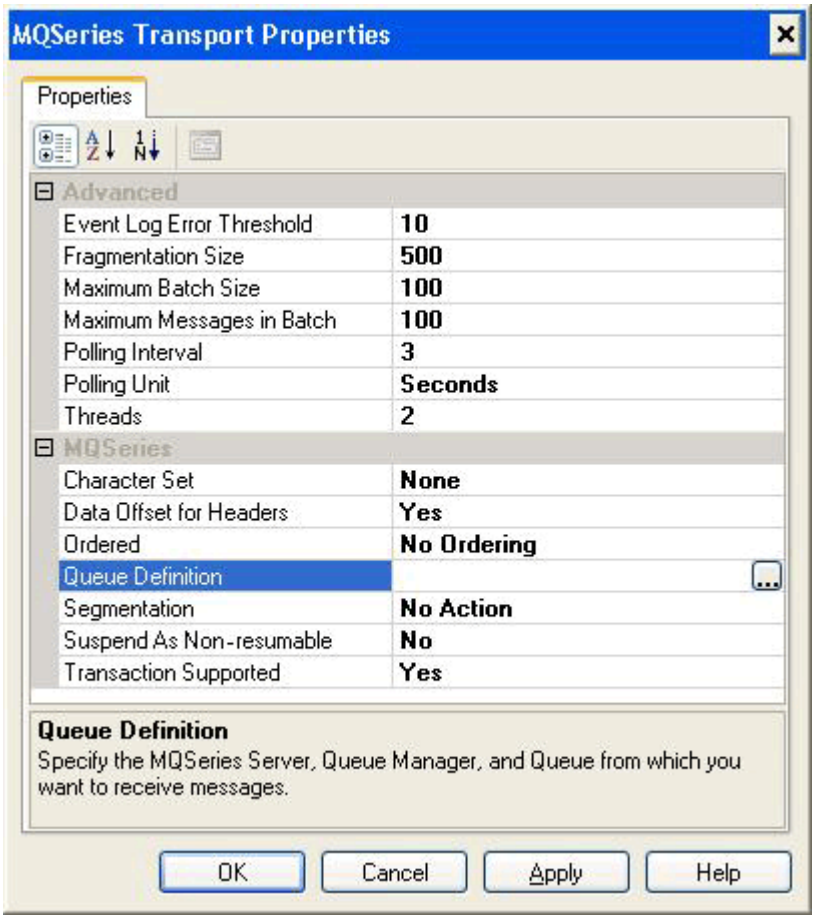
6. In the **MQSeries Transport Properties** dialog box, do the following:

Use this	To do this
Event Log Error Threshold	Determine the maximum number of errors to log. The adapter continues operating and, if the adapter recovers, it logs the event in the event log.
Fragmentation Size	Set the message chunk size in KB for messages as they are sent between MQSAgent and the adapter.
Maximum Batch Size	Determine the maximum size of a batch of messages in KB.
Maximum Messages in Batch	The maximum number of messages from 1 to 10,000 in a batch.
Polling Interval	Set the time interval from 1 to 10,000. This is the interval used by the receive component to poll the MQSeries queue. The **Polling Interval** works in combination with the hard-coded wait interval of three (3) seconds built in to the adapter. If the **Polling Interval** value is less than three seconds, the wait interval is set to the value of the **Polling Interval**.
Polling Unit	Set the unit of time for the polling interval. **Default:** Seconds
Threads	Establish the number of threads used per receive location.
Character Set	Determine the character set and whether MQSeries converts characters before sending the message to the receive location:

	• **None.** Do not convert. • **UCS-2 and UTF-16**. Convert to these character sets. MQSeries does not distinguish between them. • **UTF-8.** Convert to the UTF-8 character set. **Default:** None
Data Offset for Headers	The adapter uses values from the MQSeries headers (the MQXQH, MQIIH, and MQCIH structures) to populate corresponding values in the BizTalk Server context properties. By default, the adapter removes these MQSeries properties from the message body. Set this property to **No** to retain the properties in the message body. **Default:** Yes
Ordered	Set MQSeries to maintain the order of the messages as they are received from the MQSeries queue. Select **No Ordering** to disregard message order. Select **No Ordering with stop** to disregard message order and to disable the receive location if there is an error. Use **Order with Stop** to enable ordering. This option ends the transaction and disables the receive location if there is an error. Use **Order with Suspend** to enable ordering. This option moves the message to the suspended queue when there is an error. This value does not preserve order when there is an error, but does allow the receive location to continue receiving messages. **Default:** No Ordering
Queue Definition	Filled in with information from the **Queue Definition** dialog box.
Segmentation	Set MQSeries to assemble segmented messages or to get the message as is. Use **No Action** to read messages from the MQSeries queue without enabling segmentation. Use **Complete Message** to have MQSeries assemble segmented messages before passing them on to the adapter. **Default:** No Action
Suspend As Non Resumable	Specify whether suspended messages are marked as resumable or not. **Default:** No

Transaction Supported	The adapter begins a Microsoft Distributed Transaction Coordinator (DTC) transaction between BizTalk Server and MQSeries Server. When set to **No**, there is no guarantee of message delivery. **Default:** Yes

8. The following figure shows how you might configure the MQSeries Transport properties.



9.

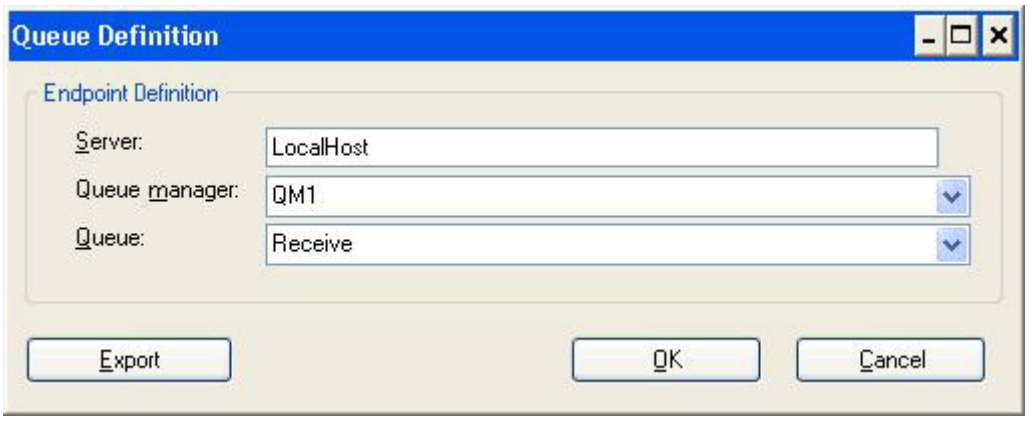
10. Click **Queue Definition**, and then click the ellipsis (**...**) button that appears to the right of the input field to complete the queue definition.

11. In the **Queue Definition** dialog box, do the following:

Use this	To do this
Server	Set the name of the server on which MQSeries for Windows is running, such as the server name or IP address.
Queue Manager	Indicate the Queue Manager to use.

Queue	Determine the MQSeries queue under the Queue Manager from which the adapter will receive messages.

12. The following figure shows values you might enter for the queue definition.



13. In the **Export** dialog box, do the following to create the queue without the IBM WebSphere MQ snap-in:

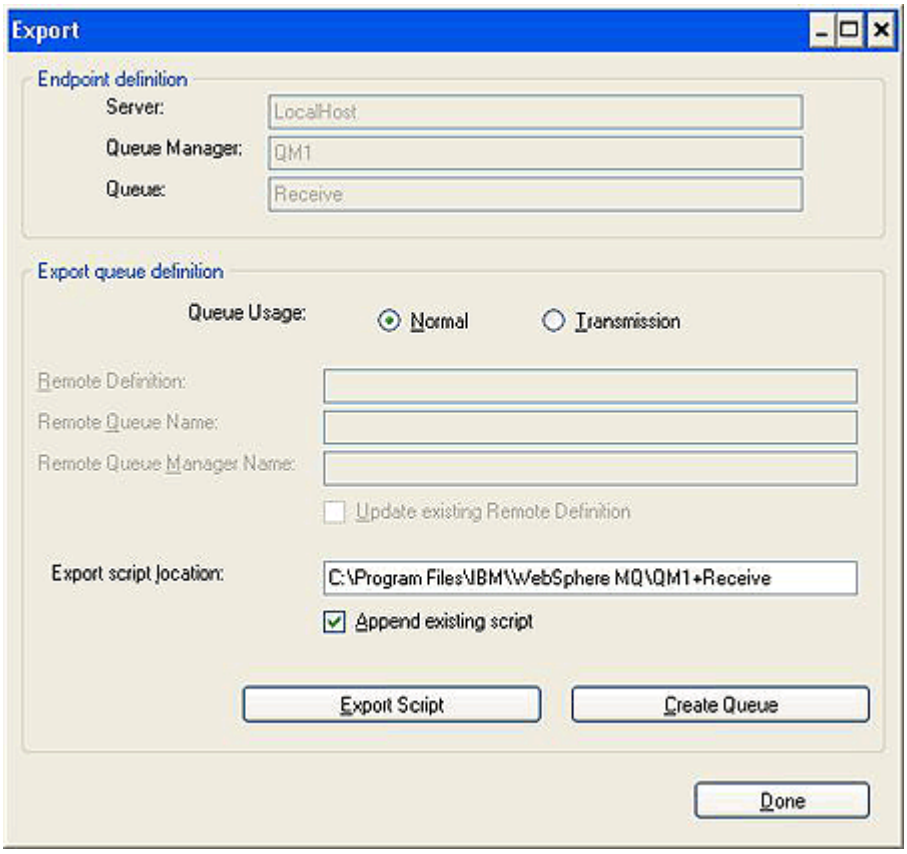
Use this	To do this
Endpoint Definition	Display the values for **Server**, **Queue Manager**, and **Queue** entered in the **Queue Definition** dialog box.
Queue Usage	Select **Normal** to create a normal MQSeries queue. Select **Transmission** if you want to create a transmission MQSeries queue. Typically, you will receive messages from a transmission queue. A transmission queue is normally associated with a remote queue definition.
Remote Definition	Set the name of the remote queue definition. Used only with **Transmission** queues.
Remote Queue Name	Determine the name of the remote queue. Used only with **Transmission** queues.
Remote Queue Manager Name	Set the name of the remote MQSeries Queue Manager. Used only with **Transmission** queues.
Export script location	Set the path and file name in which to put the exported script when you click **Export**. Use this option when you do not have permission to create the MQSeries queue directly.
Append existing script	Append the queue creation script to an existing script file.
Export Script	Export a script defining the queue to a file.

Create Queue	Create the MQSeries queue.

15. You can create the MQSeries queue immediately by clicking **Create Queue**.

    You can create a script defining the MQSeries queue. The script is useful for documenting the queue and for creating a queue on the remote server if you do not currently have the correct permissions. Selecting **Append existing script** appends the generated script to an existing script file. Otherwise, the exported file overwrites an existing file of the same name.

    The following figure shows how you might use the **Export** dialog box.



16. The **Remote Definition**, **Remote Queue Name**, and **Remote Queue Manager Name** fields are active only when you select **Transmission**.

    To create the queue on the MQSeries server with the script, you must use the MQSeries command line tool **runmqsc**. For example, if the queue definition is exported to a file named MQStoBTS.mqs, and the queue manager name is QM1, the command appears as follows:

    runmqsc QM1 <MQStoBTS.mqs

    The MQSeries administrator must have access to the script to import the definitions.

17. Click **OK** to populate the **Queue Definition** box in the **MQSeries Transport Properties** dialog box.

18. In the **MQSeries Transport Properties** dialog box, click **OK** to populate the **Address (URI)** box in the **Receive Location Properties** dialog box.

19. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and click **OK** to save settings. For information about the **Receive Locations Properties** dialog box, **To create the send port:**

1. In the BizTalk Server Administration console, create a new static send port. for more information. Configure all of the send port options and specify **MQSeries** for the **Type** option in the **Transport** section of the **General** tab.

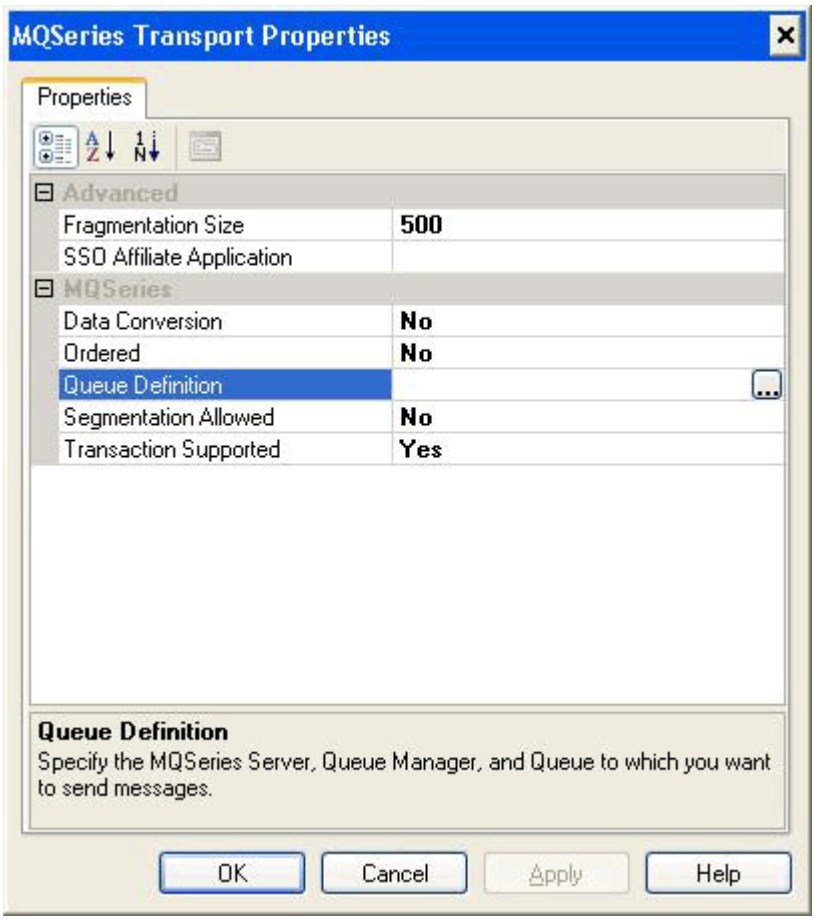
2. On the **General** tab, in the **Transport** section, click the **Configure** button next to **Type**.

3. In the **MQSeries Transport Properties** dialog box, do the following:

Property	Description
Fragmentation Size	Sets the message chunk size in KB for messages as they are sent between the adapter and MQSAgent
SSO Affiliate Application	Sets the Single Sign-On (SSO) affiliate application. The user ID and password from SSO are used for the **MQMD_UserIdentifier**, and the **MQIIH_Authenticator** (or **MQCIH_Authenticator**) property respectively. **Default:** Blank
Data Conversion	Converts the message to the ANSI code page of MQSeries for Windows server. Select **Yes** to perform this conversion from Unicode to ANSI. **Default:** No
Ordered	Sets MQSeries to maintain the order of messages as they are sent to the MQSeries queue. Select **Yes** to maintain message order. **Default:** No
Queue Definition	Populated with information from the **Queue Definition** dialog box or directly in the field.

Segmentation Allowed	Uses MQSeries Queue Manager segmentation if an individual message exceeds the MQSeries queue maximum message length. If you select **Yes**, MQSeries puts segmented messages into the queue. **Default:** No
Transaction Supported	The adapter begins a DTC transaction between BizTalk Server and MQSeries Server. When set to **No**, there is no guarantee of message delivery. **Default:** Yes

4. The following figure shows how you might configure these properties.



5. Click the ellipsis (**...**) button to the right of the **Queue Definition** box to define the queue. You can use the **Export** dialog box, just as you may have with the receive location, to create the queue immediately or to export a script defining the queue.

6. Click **OK** in each dialog box to close it and save the settings.

To enlist the send port, start the send port, and enable the receive location:

1. Right-click the send port and click **Enlist** to enlist the send port.
2. Right-click the send port and click **Start** to start the send port.
3. Right-click the receive location and click **Enable** to enable the receive location.
4. Review the event log to verify that there are no BizTalk Server errors.

How to Configure MQSeries Adapter Send and Receive Handlers

You can configure some properties for the send and receive handlers for the MQSeries adapter through the BizTalk Server Administration console. The process described in How to Configure MQSeries Adapter Receive Locations and Send Ports establishes the values for the majority of send handler properties.

To configure the send and receive handlers

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, click to expand **Platform Settings**, and then click to expand **Adapters**.
3. In the expanded adapter list, select **MQSeries**. The list of send and receive handlers that are bound to the MQSeries adapter appear in the right pane.
4. In the right pane, double-click a send or receive handler in the right pane.
5. In the **Adapter Handler Properties** dialog box, click **Properties**.
6. In the **MQSeries Transport Properties** dialog box, do the following:

Use this	To do this
Maximum Messages in Batch	Set the maximum number of messages in a batch. Applies only to the send handler.
Server	The name of the computer that is running MQSeries Server for Windows.

7. Click **OK**.

Using the MQSAgent COM+ Configuration Wizard

The MQSAgent COM+ Configuration Wizard configures the MQSAgent, the COM+ application (MQSeries component) part of the adapter. The wizard sets the application identity of the component, and the role name and users included in the role. The name of the MQSAgent COM+ component created with the MQSAgent COM+ Configuration Wizard is **MQSAgent2**.

To set the application identity

- Use the **Application Identity** page of the MQSAgent COM+ Configuration Wizard to set the application identity for the MQSAgent as follows:

Use this	To do this
Interactive User	Select this option to use the current logon account for the application identity.
Local Service	Set the application identity to a built-in service account.
Network Service	Set the application identity to a built-in account with network access.
This user	Set the application identity to the indicated user name.

To name the role and add users to it

- Use the **Name of Role** page of the MQSAgent COM+ Configuration Wizard to assign a name and users to the role as follows:

Use this	To do this
Name of role	Type the name of the role.
Users	Display the users that belong to the role.
Add	Add users to the role. These are the BizTalk Server service accounts using the adapter.

If the MQSAgent COM+ component is installed on a Windows 2000 Server computer and the MQSeries Adapter (which is installed with BizTalk Server 2006) is installed on a Windows Server 2003 computer, the MSDTC Security configuration on the Windows Server 2003 computer must be set to **No Authentication Required**.

To set the MSDTC Security configuration on the Windows Server 2003 computer to No Authentication Required

1. Click **Start**, point to **Settings**, and click **Control Panel**.
2. Double-click **Administrative Tools**.

3. Double-click **Component Services** to launch the **Component Services** management interface.

4. Expand **Component Services**, expand **Computers**, and then expand **My Computer**.

5. Right-click **My Computer** and click the **Properties** menu item.

6. In the **My Computer** dialog box, click the **MSDTC** tab and then click **Security Configuration**.

7. In the **Security Configuration** dialog box, in the **Transaction Manager Communication** section, select **No Authentication Required**. If you are prompted with a dialog box, click **Yes** to restart the MS DTC Service.

8. After the MS DTC service has restarted, click **OK** and click **OK** again to close the **My Computer** dialog box.

9. Close the **Component Services** management interface.

Silent Configuration of the MQSeries Adapter

You can use the MQSAgent COM+ Configuration Wizard from the command line to perform a silent configuration. You can also use the command-line version of the wizard to remove the MQSAgent. The wizard reads configuration information from an XML file.

In This Section

- Command-Line Configuration Wizard for the MQSeries Adapter
- XML Configuration File for the MQSeries Adapter

Command-Line Configuration Wizard for the MQSeries Adapter

The wizard has four options for installing, uninstalling, and logging actions.

Syntax

mqsconfigwiz [/u] [/i config.xml] [/l logfile] [/?]

Option	Description
/u	Uninstalls the MQSAgent.
/i *config.xml*	Installs the MQSAgent using the information in the file *config.xml*.
/l *logfile*	Logs actions to the file *logfile*.
/?	Displays a dialog box describing the command-line options.

The contents of the XML file that contains the configuration information may vary, depending on the version of Windows you are using. For more information about the configuration file format

XML Configuration File for the MQSeries Adapter

The XML configuration file read by **mqsconfigwiz** contains the same information a user enters when using the Windows version of the wizard. This information includes the application identity and the user ID and password if required, the role name, and a list of users who are part of that role.

MQSeries Adapter Properties

To access MQSeries header properties from a BizTalk orchestration, you must add a reference to the MQSeries.dll assembly to your project. This assembly is located where you installed the MQSeries adapter, for example, <*drive*:>\Program Files\Microsoft BizTalk Server 2006.

After you reference the MQSeries property schema, additional context properties are available to various BizTalk Server development tools (for example, the **Message Assignment** shape in Orchestration Designer).

The adapter automatically promotes some MQSeries properties. Your applications and custom components must avoid demoting these properties. You can promote additional properties by using custom pipeline components. The automatically promoted properties are as follows:

- BizTalk_CorrelationID
- MQMD_CorrelId
- MQMD_MsgId
- MQMD_ReplyToQ
- MQMD_ReplyToQMgr
- MQXQH_RemoteQMgrName
- MQXQH_RemoteQName
- MQXQH_MsgDesc_CorrelId
- MQXQH_MsgDesc_MsgId
- MQXQH_MsgDesc_ReplyToQ
- MQXQH_MsgDesc_ReplyToQMgr

In This Section

- Data Type Conversion of Properties
- Properties Related to BizTalk Server
- MQSeries Context Properties

Data Type Conversion of Properties

Header properties in an MQSeries message are data structures contained in the message itself. The MQSeries adapter automatically validates and converts certain values in MQSeries message headers when sending and receiving messages.

The following table describes the MQSeries data types and their validation and conversion.

MQSeries data type	Validation and conversion
MQLONG	MQSeries performs the validation. Converts to a long integer. Values that are not valid prevent the message from going to the MQSeries queue.
MQCHAR	Converts to a string.
MQBYTE	Converts to a string that contains the characters 0-9 and a-f or A-F, representing the hexadecimal value of the number.

Many of the MQSeries properties are 32-bit (4-byte) unsigned integers. Because **uint** is not a Common Language Specification (CLS)-compliant type, you must assign them to **object** types before using them in .NET methods.

Properties Related to BizTalk Server

The MQSeries adapter assigns values to some context properties that are not directly related to MQSeries but are still useful in your applications.

All the properties receive values during sending and receiving except for **BizTalk_CorrelationID**. The **BizTalk_CorrelationID** property has a value only during receiving.

Name	Type	Description
BizTalk_CorrelationID	string	Use this property to have the MQSeries server generate a correlation identifier for use with the message. For more information, see Correlating Messages Using Request-Reply .

DataConversion	string	Converts the message to the ANSI code page of MQSeries Server for Windows. On Send, if the message format is not MQFMT_STRING, there is no conversion. Select **Yes** to perform this conversion from Unicode to ANSI. **Default:** No
Ordered	string	Sets MQSeries to maintain the order of the messages as they are received from or sent to the MQSeries queue. The property has different sets of values for sending and receiving. For information about the values, see. How to Configure MQSeries Adapter Receive Locations and Send Ports . **Default:** No
SegmentationAllowed	string	Sets MQSeries to assemble segmented messages or to get the message as is. The property has different sets of values for sending and receiving. When receiving, use **No Action** to read messages from the MQSeries queue without enabling segmentation; use **Complete Message** to have MQSeries assemble segmented messages before passing them on to the adapter. **Default:** No Action When sending, select **Yes**, to have MQSeries put segmented messages in the queue. **Default:** No
SSOAffiliateApplication	string	Sets the Single Sign-On (SSO) affiliate application. You use the user ID and password from SSO for the **MQMD_UserIdentifier**, and the **MQIIH_Authenticator** (or **MQCIH_Authenticator**) property respectively. Used only when sending messages. **Default:** Blank
CompleteMessage	string	Designates whether to retrieve "complete message" when retrieving segmented messages from a queue. Set this to **Yes** to retrieve the "complete message" for segmented messages in a queue.

		Default: No
DynamicReceive	string	Designates whether to receive messages from a queue dynamically. Set this to **Yes** when receiving messages from a queue dynamically. This feature is used in conjunction with a solicit-response send port. If you specify match options (MessageID, CorrelationID, or GroupID), then only messages that correlate to the match criteria will be retrieved.
TransactionSupported	string	The adapter begins a Microsoft Distributed Transaction Coordinator (DTC) transaction between BizTalk Server and MQSeries Server. When set to **No**, there is no guarantee of message delivery. **Default:** Yes

MQSeries Context Properties

The MQSeries adapter provides a set of context properties, specific to MQSeries, for use in your applications. You can use these properties in filter expressions and in your orchestrations.

To assign MQSeries context properties to a message destined to a send port that is bound to the MQSeries adapter, use the message assignment operator and specify one of the available context properties in the MQSeries namespace.

The following is an example of setting the MQSeries **MQMD_UserIdentifier** property:

You must obtain enumerated values from the C programming language header files included with the IBM MQSeries SDK. You can find these files in the Program Files\IBM\WebSphere MQ\Tools\c\include folder. These files define the values to use when setting or reading MQSeries context property values.

Hexadecimal string values are character strings representing binary values. They do not have a prefix such as 0x. They contain digits from 0 through 9 and letters from "a" through "f" or "A" through "F". The adapter ignores white space in them.

For more information about these properties, see the IBM WebSphere MQ documentation.

The following table shows the complete set of available Message Descriptor (MQMD structure) properties and their corresponding types and values.

Name	Type	Length	Value
MQMD_AccountingToken	string	64	Hexadecimal string
MQMD_ApplIdentityData	string	32	Hexadecimal string
MQMD_ApplOriginData	string	4	String **Default:** space
MQMD_BackoutCount	unsigned int	4	Number Read only **Default:** 0
MQMD_CodedCharSetId	unsigned int	4	Number **Default:** 0
MQMD_CorrelId	string	48	Hexadecimal string
MQMD_Encoding	unsigned int	4	Number Use header file value. **Default:** 0
MQMD_Expiry	unsigned int	4	Number
MQMD_Feedback	unsigned int	4	Number Use header file value. **Default:** 0
MQMD_Format	string	8	String If set to MQXMIT, makes sure that the MQXQH properties have values.
MQMD_GroupID	string	48	Hexadecimal string
MQMD_MsgFlags	unsigned int	4	Number Use header file value. **Default:** 0
MQMD_MsgId	string	48	Hexadecimal string
MQMD_MsgSeqNumber	unsigned int	4	

MQMD_MsgType	unsigned int	4	Number Use header file value.
MQMD_Offset	unsigned int	4	
MQMD_OriginalLength	unsigned int	4	
MQMD_Persistence	unsigned int	4	Number Use header file value.
MQMD_Priority	unsigned int	4	Number
MQMD_PutApplName	string	28	String **Default:** space
MQMD_PutApplType	unsigned int	4	Number Use header file value. **Default:** 0
MQMD_PutDate	string	8	Date
MQMD_PutTime	string	8	Time
MQMD_ReplyToQ	string	48	String **Default:** space
MQMD_ReplyToQMgr	string	48	String **Default:** space
MQMD_Report	unsigned int	4	Number Use header file value.
MQMD_UserIdentifier	string	12	String Contains the user identifier when you use the **SSOAffiliateApplication** property.

When receiving messages directly from MQSeries transmission queues, the MQSeries adapter formats the transmission queue header properties (the MQXQH data structure) and places

them in their corresponding context properties. When sending messages directly to MQSeries transmission queues, the header properties are formatted and assigned values from the corresponding context properties only if the **MQMD_Format** property has a value of MQXMIT. The following table describes the properties.

Name	Type	Length	Value
MQXQH_RemoteQMgrName	string	48	String
MQXQH_RemoteQName	string	48	String

Together with the properties listed earlier in this topic, the adapter populates the following Message Descriptor values following the same rules. The adapter prefixes these property names with MQXQH_ instead of MQMD_, but otherwise they map directly to those properties defined in the Message Descriptor table:

- MQXQH_MsgDesc_AccountingToken
- MQXQH_MsgDesc_ApplIdentityData
- MQXQH_MsgDesc_ApplOriginData
- MQXQH_MsgDesc_BackoutCount
- MQXQH_MsgDesc_CodedCharSetId
- MQXQH_MsgDesc_CorrelId
- MQXQH_MsgDesc_Encoding
- MQXQH_MsgDesc_Expiry
- MQXQH_MsgDesc_Feedback
- MQXQH_MsgDesc_Format
- MQXQH_MsgDesc_MsgId
- MQXQH_MsgDesc_MsgType
- MQXQH_MsgDesc_Persistence
- MQXQH_MsgDesc_Priority
- MQXQH_MsgDesc_PutApplName
- MQXQH_MsgDesc_PutApplType
- MQXQH_MsgDesc_PutDate

- MQXQH_MsgDesc_PutTime
- MQXQH_MsgDesc_ReplyToQ
- MQXQH_MsgDesc_ReplyToQMgr
- MQXQH_MsgDesc_Report
- MQXQH_MsgDesc_UserIdentifier

There are additional MQSeries-related properties included in the property schema and available for use in filtering expressions. The following table lists these properties.

Name	Type	Length	Value
MQCIH_AbendCode	string	4	
MQCIH_ADSDescriptor	unsigned int	4	
MQCIH_AttentionId	string	4	
MQCIH_Authenticator	string	8	Set to the SSO password when you use the **SSOAffiliateApplication** property.
MQCIH_CancelCode	string	4	
MQCIH_CompCode	unsigned int	4	
MQCIH_ConversationalTask	unsigned int	4	
MQCIH_CursorPosition	unsigned int	4	
MQCIH_ErrorOffset	unsigned int	4	
MQCIH_Facility	string	16	Hexadecimal string
MQCIH_FacilityKeepTime	unsigned int	4	
MQCIH_FacilityLike	string	4	
MQCIH_Flags	unsigned int	4	

MQCIH_Format	string		
MQCIH_Function	string	4	
MQCIH_GetWaitInterval	unsigned int	4	
MQCIH_LinkType	unsigned int	4	
MQCIH_NextTransactionId	string	4	
MQCIH_OutputDataLength	unsigned int	4	
MQCIH_Reason	unsigned int	4	
MQCIH_ReplyToFormat	string		
MQCIH_ReturnCode	unsigned int	4	
MQCIH_StartCode	string	4	
MQCIH_TaskEndStatus	unsigned int	4	
MQCIH_TransactionId	string	4	
MQCIH_UOWControl	unsigned int	4	
MQIIH_Authenticator	string	8	Set to the SSO password when you use the **SSOAffiliateApplication** property.
MQIIH_CommitMode	string		
MQIIH_Flags	unsigned int	4	
MQIIH_Format	string		
MQIIH_LTermOverride	string	8	
MQIIH_MFSMapName	string	8	
MQIIH_ReplyToFormat	string		

MQIIH_SecurityScope	string		
MQIIH_TranInstanceId	string	32	Hexadecimal string
MQIIH_TranState	string		

Walkthrough: Creating a BizTalk Application That Uses the MQSeries Adapter

This section takes you through creating a simple Microsoft BizTalk Server 2006 application that uses the MQSeries adapter.

The application is a simple content-based routing application using only a receive location and a send port. The receive location reads from an IBM WebSphere MQ queue. The send port takes the message from the receive location and sends it to a different IBM WebSphere MQ queue.

To create the application, you have to create the IBM WebSphere MQ queues, set up the BizTalk Server receive location and send port, start the send port and enable the receive location, and put a test message in the queue.

If you have the required permissions to the IBM WebSphere MQ installation, you can create the IBM WebSphere MQ queues through the adapter dialog boxes, and can skip the next procedure. If you do not have such access, you can create the queues using the IBM WebSphere MQ, client components for Windows platforms Explorer. To create the queues through the IBM WebSphere MQ Explorer snap-in, perform the following procedure.

To create the IBM WebSphere MQ queues through the IBM WebSphere MQ Explorer

1. Click **Start**, point to **Programs**, point to **IBM WebSphere MQ**, and then click **WebSphere MQ Explorer**.

2. Double-click **Queue Managers**, and then double-click the default queue manager. The default queue manager is typically named **QM_**<*machine_name*> where *machine_name* is the name of your computer.

3. Right-click **Queues**, point to **New**, and then click **Local Queue**.

4. In **Create Local Queue** dialog box, in **Queue Name**, type **BTStoMQS**, and then click **OK**.

5. Right-click **Queues**, point to **New**, and then click **Local Queue**.

6. In **Create Local Queue** dialog box, in **Queue Name**, type **MQStoBTS**, and then click **OK**.

The next steps create the receive location and the send port, and start the send port and enable the receive location. They also create the IBM WebSphere MQ queues.

To create the receive location and the MQSeries queue

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the default application (**BizTalk Application 1** by default).

2. Right-click the **Receive Ports** node, click **New**, and select **One-Way Port**.

3. In the **Receive Port Properties** dialog box, in the **Name** box, type **MQStoBTS**.

4. In the left pane, click **Receive Locations**, and in the right pane, click **New**.

5. In the **Receive Location Properties** dialog box, in the **Name** box, type **MQStoBTS**.

6. Select **MQSeries** from the drop-down list next to the **Type** option.

7. In the **Transport** section, click **Configure**.

8. In the **MQSeries Transport Properties** dialog box, in the **Polling Interval** box, type **1**.

9. In the **Queue Definition** box, click the ellipsis (**...**) button.

10. In the **Queue Definition** dialog box, in the **Server Name** box, type your computer name.

11. In the **Queue Manager** box, select the default queue manager.

12. In the **Queue** box, type **MQStoBTS**, and then click **Export**.

13. In the **Export** dialog box, click **Create Queue**,and then click **OK** and **OK** again to return to the **Receive Location Properties** dialog box.

14. In the **Receive Handler** box, select **BizTalkServerApplication**.

15. In the **Receive Pipeline** box, select **PassThruReceive**.

16. Click **OK** to apply changes.

To create the send port and the MQSeries queue

1. Right-click **Send Ports**, click **New**, and select **Static One-way Send Port**.

2. In the **Send Port Properties** dialog box, in the **Name** box, type **BTStoMQS.**

3. Select **MQSeries** from the drop-down list next to the **Type** option.

4. In the **Transport** section, click **Configure**.

5. In the **MQSeries Transport Properties** dialog box, in the **Queue Definition** box, click the ellipsis (**...**) button.

6. In the **Queue Definition** dialog box, in the **Server Name** box, type your computer name.

7. In the **Queue Manager** box, select the default queue manager.

8. In the **Queue** box, type **BTStoMQS**, and then click **Export**.

9. In the **Export** dialog box, click **Create Queue**, and then click **OK** and **OK** again to return to the **Send Port Properties** dialog box.

10. In the **Send Pipeline** box, select **PassThruTransmit**.

11. Click to select **Filters** in the left pane, and then configure filter options in the right pane.

12. In the **Property** drop-down list, select **BTS.ReceivePortName**.

13. In the **Value** box, type **MSQtoBTS**.

14. Click **OK** to apply changes.

To enable the receive location and start the send port

1. Right-click the **MQStoBTS** receive location, and then click **Enable**.

2. Right-click the **BTStoMQS** send port, and then click **Start**.

The next step is to test the application by sending a test message to the receive queue.

To test the application

1. Click **Start**, point to **Programs**, point to **IBM WebSphere MQ**, and then click **WebSphere MQ Explorer**.

2. Right-click **MQStoBTS**, and then click **Put Test Message**.

3. In the **Message Data** box, type a test message. Click **OK**.

After you enter the data, the **Current Depth** for the **MQStoBTS** queue is one (1). When the application processes the message, the count returns to zero (0) and the **Current Depth** for **BTStoMQS** becomes one (1). You can also view the content of the message.

To view the message

1. Double-click the **BTStoMQS** queue.

2. Double-click the message, and then select the **Data** sheet. You can view the text of the message in the **Message Data** box.

3. Click **OK**.

MQSeries Adapter Batching and Transaction Handling

The MQSeries adapter stops a transaction only if it does not receive all the data. The boundaries of a transaction for the adapter are the adapter endpoint (MQSeries queue on the MQSeries Server) and the MessageBox database.

If the BizTalk application invalidates a message, the adapter moves the message to the suspended queue. However, because it is still a valid transaction from the point of view of the adapter (the adapter received all the data), the adapter commits the transaction.

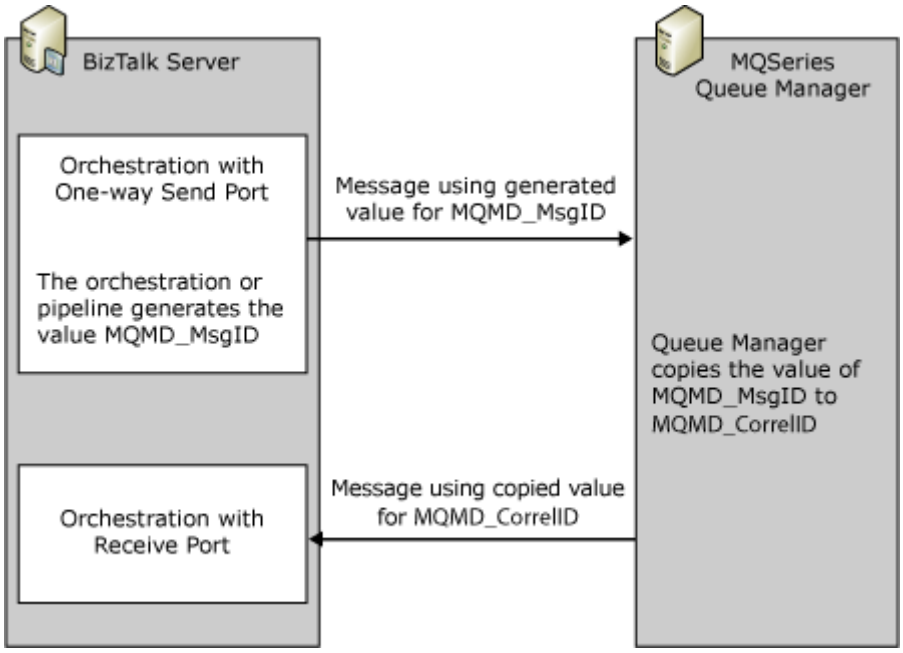
You can control batching and transaction handling by setting properties when configuring the adapter.

Correlating Messages Using Request-Reply

There are two ways to correlate messages in BizTalk orchestrations for IBM WebSphere MQ, server component for Windows platforms request-reply scenarios. The first is to supply the correlation identifier by setting both the MessageID (**MQMD_MSGID**) and the CorrelationID (**MQMD_CorrelId**) to the same value. The second is to use the **BizTalk_CorrelationId** context property.

Setting MQMD_MsgId and MQMD_CorrelId to the Same value

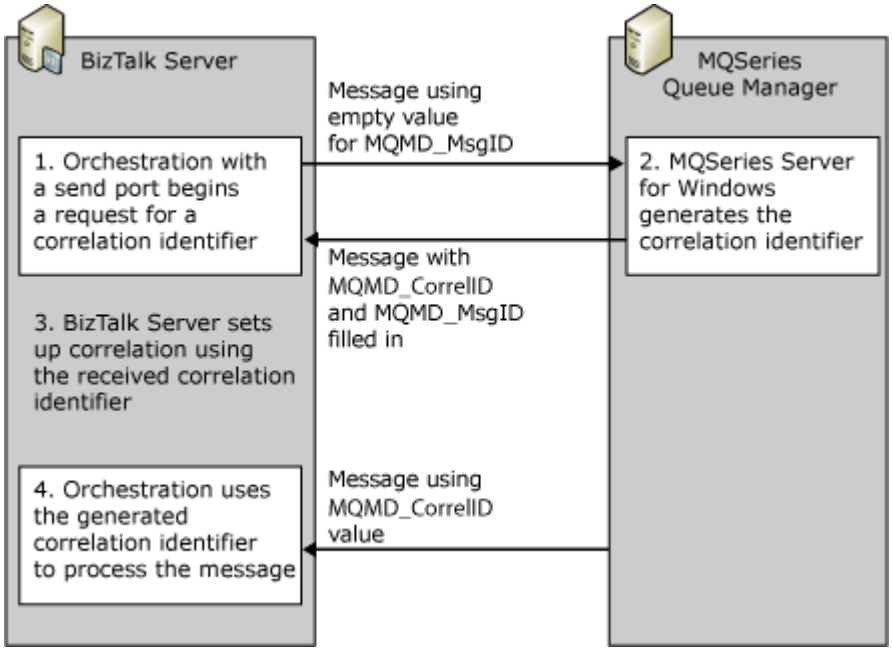
When sending the message to an IBM WebSphere MQ Queue Manager, you can set the message identifier (**MQMD_MSGID**) and the correlation identifier (**MQMD_CorrelId**) to the same value in the outgoing message. The IBM WebSphere MQ Queue Manager copies the MessageID to the CorrelationID for the reply message. The following figure shows the process.



You can initialize the correlation sets for the outgoing message and follow the correlation sets for the incoming message using the value of **MQMD_CorrelId**.

Using the MQSeries.BizTalk_CorrelationId Context Property

Instead of setting the MessageID and CorrelationID to the same value in the outgoing message, you can use the **BizTalk_CorrelationID** context property with a solicit-response send port of the MQSeries adapter. The following figure shows this process.



To use identifiers provided by IBM WebSphere MQ Server for correlations in your BizTalk orchestration, BizTalk Server must first obtain the identifier. Your application does this through a solicit-response request. BizTalk Server sends a solicit-response request to the IBM WebSphere MQ Server by using the MQSeries adapter. In return, it receives a response with the message identifier (**MQMD_MSGId**) and the correlation identifier (**MQMD_CorrelId**).

For the outgoing message in a solicit-response send port, the adapter copies the **MQMD_MSGID** generated by IBM WebSphere MQ Server to the **MQSeries.BizTalk_CorrelationId** context property.

When receiving messages, the adapter copies the **MQMD_CorrelId** to the **MQSeries.BizTalk_CorrelationId**. In this case, using correlation sets, you can initialize the correlation sets for the outgoing message and follow the correlation sets for the incoming message using the **MQSeries.BizTalk_CorrelationId**.

MQSeries Adapter Custom Headers

Because of the header structures used in MQSeries messages, you must manage any custom headers you want to use. Custom headers must be part of the message body to avoid interfering with the processing of the MQSeries headers. Make sure that you avoid demoting any one of the automatically promoted properties. For more information about automatically promoted properties,

In turn, the incorporation of custom headers in the message body requires additional processing. One solution is to handle the custom headers in the pipelines of the application. A receive pipeline extracts the custom header information and promotes the information as context properties. Similarly, the send pipeline takes the context properties corresponding to the custom header and demotes the properties in the body of the message.

Analyzing MQSeries Adapter Errors with the Trace Tools

You use the trace tools to analyze messaging failures when you run your application. With the MQSeries adapter you must use two tools, one for the adapter and your BizTalk application (trace.cmd), and the other for the MQSAgent (MQSTrace.cmd). Both tools use tracelog.exe. You have to install tracelog.exe if you do not already have it.

With both trace.cmd and MQSTrace.cmd you have to set an option (**-tools**) so that these tools can find the tracelog.exe file.

Install the Trace Utility

To install the BizTalk Adapter Trace Utility, follow these steps:

1. To download the Tracelog.exe file, visit the Microsoft Platform SDK download Web site at http://go.microsoft.com/fwlink/?LinkId=21975.

2. Start the Platform SDK Web installation program by clicking the link for the **PSDK-x86.exe** file at the bottom of the Web page.

3. When you are prompted, choose the option for a custom installation.

4. In the **Custom Installation** dialog box, click to clear all the available features.

5. Expand the **Microsoft Windows Core SDK** feature, and then expand the **Tools** feature.

6. Choose the **Tools (Intel 64-bit)** feature, and then click **Will be installed on local hard drive**.

7. Click **Next**, and then click **Next** again to start the installation.

8. Locate the *drive*:*MicrosoftPlatformSDKInstallationFolder**bin* folder, and then copy the Tracelog.exe file to the Microsoft BizTalk Server installation folder. The BizTalk Server installation folder also contains the Trace.cmd file.

Enable the Trace Utility

To enable the BizTalk Adapter Trace Utility in BizTalk Server, follow these steps:

1. Move to the directory that contains trace.cmd. The default location is the Microsoft BizTalk Server 2006 directory.

2. Type the following command, substituting the directory that contains the tracelog.exe file on your computer for the directory in quotes, and then press ENTER:

   trace -tools "c:\Program Files\Microsoft SDK\Bin"

3. Move to the directory that contains MQSTrace.cmd.

4. Type the following command, substituting the directory that contains the tracelog.exe file on your computer for the directory in quotes, and then press ENTER:

   MQSTrace -tools "c:\Program Files\Microsoft SDK\Bin"

Run the Trace Utility

To run the BizTalk Adapter Trace Utility, follow these steps:

1. At the command prompt, type **trace.cmd -start -high**, and then press ENTER.

2. Run your failure scenario.

3. At the command prompt, type **trace.cmd -stop**, and then press ENTER.

4. The bts2006.bin file contains the output of the trace tool. Contact Microsoft Product Support Services for analysis.

To run the MQSAgent Trace Utility, follow these steps:

1. At the command prompt, type **MQSTrace.cmd -start -high**, and then press ENTER.

2. Run your failure scenario.

3. At the command prompt, type **MQSTrace.cmd -stop**.

4. The MQSAdapterTrace.bin file contains the output of the trace tool. Contact Microsoft Product Support Services for analysis

Ordered Delivery of Messages with the MQSeries Adapter

BizTalk Server 2006 provides an **Ordered Delivery** option for static send ports. Setting the **Ordered Delivery** option on a send port to **True** ensures that BizTalk Server delivers messages to the send port in the same order that they are published to the BizTalk MessageBox database. To provide end-to-end ordered delivery the following conditions must be met:

- Messages must be received with an adapter that preserves the order of the messages when submitting them to BizTalk Server. For example, when receiving messages with the MQSeries receive adapter, the receive location should be configured with the option **Order with Stop** or **Order with Suspend**.

- You must subscribe to these messages with a send port that has the **Ordered Delivery** option to **True**.

- If an orchestration is used to process the messages, only a single instance of the orchestration should be used, the orchestration should be configured to use a sequential convoy, and the **Ordered Delivery** property of the orchestration's receive port should be set to **True**.

Using the MQSeries Adapter for Ordered Delivery of Messages

The MQSeries receive adapter provides support for preserving the order of messages when submitting them to BizTalk Server. End-to-end ordered delivery of messages through BizTalk Server can be achieved when receiving messages with the MQSeries adapter if the messages are processed by a send port that is configured with the **Ordered Delivery** option set to **True**.

MSMQ Adapter

The MSMQ adapter lets you use Microsoft Message Queuing 2.0 and Message Queuing 3.0 from BizTalk Server 2006. Integrating Message Queuing technology with BizTalk Server enables applications that are running at different times to communicate across heterogeneous networks and systems that may be temporarily offline.

The MSMQ adapter features include the following:

- Improved performance

- Support for multithreaded operations
- Support for Message Queuing 3.0 features, including support for HTTP or HTTPS transports
- Support for processing messages larger than 4 MB

To participate in BizTalk Server 2006 online community discussions, go to http://go.microsoft.com/fwlink/?LinkId=41609.

In This Section

- What Is the MSMQ Adapter?
- Configuring the MSMQ Adapter
- Reliable Messaging with the MSMQ Adapter
- Analyzing MSMQ Adapter Errors with the Trace Tool
- MSMQ Adapter Property Schema and Properties
- Ordered Delivery of Messages with the MSMQ Adapter

What Is the MSMQ Adapter?

With the BizTalk Server 2006 Adapter for MSMQ (the MSMQ adapter), you can send and receive messages to Microsoft Message Queuing (also known as MSMQ) queues using Microsoft BizTalk Server 2006. The MSMQ adapter supports both Message Queuing 2.0 and Message Queuing 3.0. The adapter works with transactional and non-transactional, public and private, and local and remote queues. Additionally, the MSMQ adapter provides large (greater than 4 MB) message support and gives you access to Message Queuing features such as messaging over HTTP and multi-cast messaging.

The BizTalk Message Queuing adapter (MSMQT) and the BizTalk MSMQ adapter offer different features. The following table highlights some of the differences between the two adapters.

MSMQT adapter	MSMQ adapter
Delivers messages in order	Can be configured to deliver messages in order
Uses only Message Queuing 2.0	Uses either Message Queuing 2.0 or 3.0
Provides large message support by streaming directly to the BizTalk MessageBox database (not as memory intensive as the MSMQ adapter)	Provides large message support by breaking the message into parts, accumulating the parts in memory, and delivering the parts in order to the destination (more memory intensive than MSMQT)
Primarily provided for backward compatibility for existing BizTalk Server	Provides better performance than MSMQT

2004 solutions built around MSMQT Does not require intermediate storage of MSMQ queues. Messages are sent directly to the MessageBox database.	Enables other non-BizTalk applications to use MSMQ services at the same time on the same computer Requires intermediate storage of MSMQ queues. Inbound messages are written to the MSMQ queue and then picked up from the MSMQ queue by the MSMQ adapter.

If you have upgraded your BizTalk Server installation to BizTalk Server 2006, the BizTalk Server 2004 Adapter for MSMQ is not removed. Because BizTalk Server 2006 installs a new version of the MSMQ adapter, you can safely uninstall the BizTalk Server 2004 Adapter for MSMQ and then manually remove the directory structure for this version of the adapter. By default, the BizTalk Server 2004 Adapter for MSMQ is installed into the C:\Program Files\Microsoft BizTalk 2004 Adapter for MSMQ\ directory.

In This Section

- MSMQ Adapter Architecture

MSMQ Adapter Architecture

The MSMQ adapter lets you take advantage of Microsoft Message Queuing (also known as MSMQ) features that are otherwise unavailable in BizTalk Server. The MSMQ adapter is also multithreaded and provides better performance than MSMQT.

Adapter Structure

The MSMQ adapter has the same structure as other BizTalk adapters and uses the Adapter Framework. It is made up of a design-time component and a run-time component. The run-time component, in turn, contains the elements that implement the message transport.

The design-time component lets you configure the adapter properties for sending and receiving.

The run-time component can send messages to a queue defined at design time or receive messages from a designated queue. The adapter runtime runs in the same process as the BizTalk Server application and does not run in an isolated host.

All message handling relies on the local Message Queuing service, even for remote queues. For remote queues, the adapter hands messages off to the local Message Queuing service. It, in turn, sends the messages to the remote queue.

For a complete list of send and receive configuration properties

Configuring the MSMQ Adapter

This section includes information about configuring the MSMQ adapter. You can configure the MSMQ adapter for both receive locations and send ports.

In This Section

- How to Configure an MSMQ Receive Handler
- How to Configure an MSMQ Receive Location
- How to Configure an MSMQ Send Handler
- How to Configure an MSMQ Send Port
- Configuring the MSMQ Adapter Properties
- How to Manage Multiple Receive Locations
- Message Queuing Queues
- Optimizing Performance of the MSMQ Adapter
- Sending and Receiving Large Messages
- Setting Up the MSMQT and MSMQ Adapters on One Computer

How to Configure an MSMQ Receive Handler

Use the following procedure to change the host with which the MSMQ receive handler is associated.

To change the host with which the MSMQ receive handler is associated

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.
2. In the expanded adapter list, click **MSMQ**, in the right pane, right-click the receive handler that you want to configure, and then click **Properties**.
3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.
4. Click **OK**.

How to Configure an MSMQ Receive Location

You can set MSMQ receive location adapter variables in the BizTalk Server Administration console. If properties are not set in the receive location, the default receive handler values set in the BizTalk Server Administration console are used.

To configure variables for an MSMQ receive location

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application you want to create a receive location in.

2. In the BizTalk Server Administration console, in the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

3. In the **Receive Port Properties** dialog box, in the left pane, select **Receive Locations**, and then in the right pane, double-click an existing receive location or click **New** to create a new receive location.

4. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **MSMQ** from the drop-down list, and then click **Configure**.

5. In the **MSMQ Transport Properties** dialog box, do the following:

Use this	To do this	Date type	Default value
Password	Set a password to use for a remote queue.	String	Blank
User Name	Determine the user name to use, in combination with the password, for access to a remote queue. You cannot use the local user of the remote computer for the user name.	String	Blank
Batch Size	Configure the batch size. The MSMQ adapter submits messages to the MessageBox database in batches. The default batch size is 20, and the minimum batch size is 1.	Int	20
On Failure	Specify how the adapter should respond to an error. Set this property to one of the following values: • **Stop.** Stop receiving messages through this receive location if an error condition occurs. • **Suspend(non-resumable).** Suspend messages and mark as non-resumable. • **Suspend(resumable).** Suspend messages and mark as resumable.	String	Suspend(resumable)

Ordered Processing	Set this property to **True** or **False**. This indicates whether to process messages serially. Setting the property to **True** will accommodate ordered message delivery when used in conjunction with a BizTalk messaging or orchestration send port that has the **Ordered Delivery** option set to **True**. Setting this property to **True** also optimizes resource usage when handling large messages by making the adapter single-threaded..	Boolean	False
Queue	Type a valid queue path. Depending on the queue path you specify, the system performs the appropriate validations.	String	Blank
Transactional	Set this property to **True** or **False**.	Boolean	False

7. Click **OK**.

8. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and click **OK** to save settings. For information about the **Receive Locations Properties** dialog box,

How to Configure an MSMQ Send Handler

Use the following procedure to change the global variables for an MSMQ send handler.

To change global variables for an MSMQ send handler by using the BizTalk Server Administration console

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **MSMQ**, in the right pane, right-click the send handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the send handler will be associated, and then click **Properties**.

4. In the **MSMQ Transport Properties** dialog box, enter a value for **Batch Size**.

The MSMQ send handler will send messages to destination queues using the specified **Batch Size** parameter. The default **Batch Size** value is 5.

5. Click **OK** and click **OK** again to close the **Adapter Handler Properties** dialog box.

How to Configure an MSMQ Send Port

You can set MSMQ send port adapter variables in the BizTalk Server Administration console. If properties are not set for the send port, the default send handler values set in the BizTalk Server Administration console are used.

To configure variables for an MSMQ send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. for more information. Configure all of the send port options. On the **General** tab, in the **Transport** section, specify **MSMQ** for the **Type** option.

2. On the **General** tab, in the **Transport** section, click the **Configure** button next to **Type**.

3. In the **MSMQ Transport Properties** dialog box, do the following:

Use this property	To do this	Data type	Default value
Password	Specify the password for a remote queue. Use with **User Name**.	String	Blank
User Name	Specify the user name for a remote queue. Use with **Password**. You cannot use the local user of the remote computer for the user name.	String	Blank
Acknowledgement Type	Specify the type of acknowledgement message for Message Queuing to return to the sending application. You can select more than one acknowledgment type. Any of the acknowledgement types in the **System.Messaging.AcknowledgeTypes** enumeration are available.	String	None
Administration Queue	Specify the queue name that receives the acknowledgement message.	String	Blank
Body Type	Specify the message body type in MSMQ. Valid values are members of the .NET **VarEnum** enumeration.	Int	8209
Certificate	Specify the thumbprint of the certificate to use for	String	Blank

Thumbprint	message authentication. Use this property in combination with the **Use Authentication** property to verify the message. Use the **User Name** and **Password** properties to gain access to queues.		
Destination Queue	Specify the destination queue. For more information about queues,	String	Blank
Encryption Algorithm	Select **RC2**, **RC4**, or **None** for the encryption algorithm.	Enum	None
Maximum Message Size (in kilobytes)	Specify the maximum message size for messages that you send to the specified queue.	UnsignedInt	1024
Message Priority	Set the message priority.	Enum	Normal
Recoverable	Specify whether to guarantee the recoverability of a message.	Boolean	False
Support Segmentation	Set this Boolean property value to **True** to segment messages larger than 4 MB.	Boolean	False
Timeout	Specify the maximum time to wait for the messages to reach the destination queue. Applies only when you use transactions.	Int	0
Timeout Unit	Set the unit to use for the **Timeout** property. Select **Days**, **Hours**, **Minutes**, or **Seconds**.	Enum	Days
Transactional	Set this value to **True** to send messages if you use transactions.	Boolean	False
Use Authentication	Set this Boolean property value to **True** to control authentication. Use this property in combination with the **Certificate Thumbprint** property to verify the message. Use the **User Name** and **Password** properties to gain access to queues.	Boolean	False
Use Dead Letter Queue	Set this value to **True** to send messages to the dead letter queue if a failure occurs.	Boolean	True
Use Journal Queue	Set this value to **True** to save a copy of the message whenever the message is processed.	Boolean	False

4. Click **OK** and **OK** again to save settings.

Configuring the MSMQ Adapter Properties

This section is a complete list of the configuration properties for the MSMQ adapter. These properties determine the behavior of the MSMQ adapter when sending or receiving. The properties are marked to indicate whether they apply to sending, receiving, or both.

Acknowledgement Type

Send only. The **Acknowledgement Type** property specifies the type of acknowledgement message that the sending application requests. This property is an enumeration of the types available in the **System.Messaging.AcknowledgeTypes** enumeration. For more information, see "AcknowledgeTypes Enumeration" in .NET Framework Class Library Help.

You can set more than one value for this property. Selecting **None** clears all other options.

Default value: **None**.

Administration Queue

Send only. This is the name of the queue that receives the acknowledgement message. The MSMQ adapter does not validate the **Acknowledgement Type** against a value specified in the Administration queue.

The adapter performs Queue path syntax validation on this field. For more information, see "Queue" later in this topic.

Certificate Thumbprint

Send only. The **Certificate Thumbprint** send port property contains the thumbprint of the client certificate. You can enter the thumbprint of the certificate that you want to use for message authentication purposes.

The MSMQ adapter does not validate the **Certificate Thumbprint** property value and the **Use Authentication** value interrelationship during design time. If the **Use Authentication** value is set to **False**, the adapter does not use the **Certificate Thumbprint** property.

At run time, the send adapter tries to retrieve a certificate from the personal certificate store based on the thumbprint. If the MSMQ adapter finds the certificate, then the certificate data (byte array) is set in the **System.Messaging.Message.SenderCertificate** property.

Destination Queue

Send only. Designates the Message Queuing queue to receive messages. The string may contain information about the path of the queue, its format, and a descriptive text label. For more information about naming conventions for the queue path,.

Encryption Algorithm

Send only. The **Encryption Algorithm** options include **RC2**, **RC4**, and **None**. If you select **None**, the adapter does not encrypt messages. For message encryption, the adapter sets the message property **Message.UseEncryption**, and sets the user-defined algorithm in the message property **Message.EncryptionAlgorithm**. For more information, see "Message.UseEncryption Property" and "Message.EncryptionAlgorithm Property" in .NET Framework Class Library Help.

Default value: **None**.

Maximum Message Size (in kilobytes)

Send only. Specify the maximum message size for messages that you send to the specified queue.

Default value: **1024**.

Message Priority

Send only. Sets the message priority using the values defined in the **System.Messaging.MessagePriority** enumeration. For more information, see "MessagePriority Enumeration" in .NET Framework Class Library Help.

Default value: **Normal**.

Password

Send and receive. Specify the password for the remote queue used for sending or receiving messages. Use with **User Name**.

Queue

Receive only. Designates the Message Queuing queue. The string may contain information about the path of the queue, its format, and a descriptive text label. For more information about naming conventions for the queue path,

Send only. The **Recoverable** property indicates whether the delivery of a message is guaranteed even if a system fails while the message is in route to the destination queue.

When you set this value to **False**, the message is kept in volatile memory on each computer in route to the destination queue. If the system fails, the message is lost.

When you set this value to **True**, the message is written to disk on every computer in route to the destination queue. If the system fails, the message is not lost.

Serial Processing

Receive only. Set this property to **True** to process messages serially. Setting this property to **True** optimizes resource use when handling large messages. For more information,

Support Segmentation

Send only. The MSMQ adapter supports sending and receiving large messages (messages larger than 4 MB) through message segmentation. **Support Segmentation** is a Boolean property that you configure on the send port. If you set this property to **False** and the message is larger than 4 MB, then an exception occurs, and the adapter rejects the message. The adapter uses the Message Queuing segmentation facilities.

Time-Out

Send only. The **TimeOut** property specifies the time for the message to reach the destination queue before a time-out occurs. This property applies only to the message and does not apply to the acknowledgement sent as a response to a solicit-response request. The adapter for MSMQ only uses the **TimeOut** property value in transactional send and receive.

Default value: **4**.

Timeout Unit

Send only. Specifies the units for the **TimeOut** property. You can set the property to **Days**, **Hours**, **Minutes**, or **Seconds**.

Default value: **Days**.

Transactional

Send and receive. You can set the **Transactional** property to **True** or **False** on the send port and the receive location. The following information describes the behavior for transactional and non-transactional send ports and receive locations:

- **Transactional send.** The MSMQ adapter receives messages from BizTalk Server and groups the messages until the batch reaches the batch size set in Configuration. The batch of messages is sent to the destination. If the batch submission fails, then the adapter tries to resubmit the batch. The BizTalk Server batch handler handles a resubmit failure.

- **Non-transactional send.** The MSMQ adapter receives and accumulates messages as in transactional send. After the adapter has a batch of messages, it tries to submit the batch, but not in a transaction. If a message submission fails, then the adapter moves to the next batch.

- **Transactional receive.** The MSMQ adapter receives transactional Message Queuing messages and fills the BizTalk Server batch. After the adapter has a batch of messages, it tries to submit the batch in a single transaction. If the batch submission fails, then the adapter tries to put the messages in the BizTalk Server suspended messages queue in a

single transaction. If this fails, then the messages are rolled back into the Message Queuing queue.

- **Non-transactional receive.** The MSMQ adapter receives non-transactional Message Queuing messages and fills the BizTalk Server batch. After the adapter has a batch of messages, it tries to submit the batch, but not in a transaction. If a message submission fails, then the adapter tries to put that message in the BizTalk Server suspended messages queue. If this fails, then the message is lost.

Use Authentication

Send only. You can control authentication based on the value set for the **Use Authentication** property on the message.

If the value set is **False** and the queue accepts only authenticated messages, then the message is rejected when it reaches the queue. The Queue Manager does not maintain rejected messages in any system queue, but discards them.

If the value set is **True**, then the adapter uses the internal certificate property by default, unless you explicitly specify an external certificate.

Use Dead Letter Queue

Send only. If a transmission failure occurs, you can save messages in this queue by setting this property to **True**. The adapter saves both Transactional Dead Letter and the Dead Letter Queue (Non-Transactional) messages.

Default value: **True**.

Use Journal Queue

Send only. When **True**, the MSMQ adapter saves copies of messages in the journal queue when the messages are processed.

User Name

Send and receive. Use to specify the user name for the remote queue for sending or receiving messages. Use with **Password**. You cannot use the local user of the remote computer for the user name.

How to Manage Multiple Receive Locations

To increase performance, the MSMQ adapter is multithreaded. If you have many receive locations, there may not be enough threads available for all the receive locations. This prevents some of the receive locations from picking up messages. There are three ways to solve this problem:

- Add BizTalk Hosts to your computer and divide the receive locations among the hosts. Adding hosts makes more threads available for the receive locations.

- Set the **Serial Processing** property to **True** on each receive location. Setting the property to **True** assigns a single thread to each receive location. This leaves more threads available in the pool. However, this may also cause a decrease in performance.

- Modify the registry to increase the number of available threads. For information about editing the registry, see the following procedure.

To add threads to a BizTalk Server installation

1. Stop the BizTalk Host.

2. Click **Start**, click **Run**, type **regedit.exe**, and then click **OK** to start Registry Editor.

3. Navigate to **[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BTSSvc***guid***]** where *guid* is a GUID unique to each installation of BizTalk Server.

4. Under the **CLR Hosting** key, create the following DWORD entry with the indicated value.

DWORD entry	Value
MaxWorkerThreads	75

5. Exit Registry Editor.

6. Restart the BizTalk Host.

Message Queuing Queues

This section describes how to specify Microsoft Message Queuing (also known as MSMQ) queues when you use the MSMQ adapter. It describes the conventions for specifying paths and also describes the role that format names play in translating paths into queue designations.

Queue Path Naming Conventions

If the queue name refers to a path, use the naming conventions in the following table.

Queue type	Syntax for path
Public queue	*Computername*\QueueName
Private queue	*Computername*\Private$\QueueName
Journal queue	*Computername*\QueueName\Journal$
Computer journal queue	*Computername*\Journal$

Computer dead-letter queue	*Computername*\Deadletter$
Computer transaction dead-letter queue	*Computername*\XactDeadletter$

If the queue name refers to a format name, it takes the form of a string that indicates whether a queue is public or private, followed by a generated GUID for the queue and other identifiers as needed. Use the naming conventions in the following table.

Format type	**Syntax for format name**
Public	*FormatName*:Public=QueueGUID
Direct	*FormatName*:DIRECT=SPX:NetworkNumber:HostNumber\QueueName *FormatName*: DIRECT=TCP:IPAddress\QueueName *FormatName*: DIRECT=OS:ComputerName\QueueName

If the send port queue path is a distribution list, then the queue path syntax is:

DL=DistributionListGUID

If the send or receive queue path is an HTTP or HTTPS URL, then the syntax is:

FormatName:DIRECT=http://<client name>/msmq/<queue name>

FormatName:DIRECT=https://<client name>/msmq/<queue name>

If the queue name refers to a descriptive text label that the administrator specified for the queue, then the syntax of the queue path referring to this label is:

LABEL:MyQueue

Role of the Format Name

Message Queuing uses the format name to identify a queue and to determine how to access it. Message Queuing assigns the format name to the queue.

When you specify a queue using the path name syntax, for example myMachine\myQueue, Message Queuing looks up the path to find the associated format name. Message Queuing then uses that format name to access the queue. When you specify the format name, Message Queuing uses the format name you use.

For more information about format names, see "MessageQueue.FormatName Property" in .NET Framework Class Library Help.

Troubleshooting Queue Paths

- An exception occurs if the syntax of the provided queue path does not match one of the formats described earlier in "Queue Path Naming Conventions."

- The following are not valid characters for computer names in the queue path:

  \ ; , + "

  An exception occurs if the computer name is a number. For example: 234\private$\queue.

- For a computer dead-letter queue, computer journal queue, and computer transaction dead-letter queue, an exception occurs if the user specifies any one of the system queues as the destination queue for send.

- **System.Messaging.MessageQueue.Exists** does not work for remote queues. For more information, see "MessageQueue.Exists Method" in .NET Framework Class Library Help.

Optimizing Performance of the MSMQ Adapter

Optimization of the MSMQ adapter differs between the send and receive sides. You control optimization on the receive side by setting a property on the receive location. On the send side, you can control optimization by using an orchestration.

Receive Optimization

On the receive side, you can have the adapter use a single execution thread. Whether the adapter uses a single thread or multiple threads depends on the setting of the **Ordered Processing** property on the receive location, as follows:

- When the property is **True**, the adapter operates on a single thread. This limits the adapter to one message at a time and conserves memory. Notice that this effectively sets **Batch Size** to one (1), regardless of the value assigned to it in the property sheet.

- When **Ordered Processing** is **False**, the adapter runs multiple threads and can process multiple messages at a time, therefore increasing performance.

You must set **Ordered Processing** to **True** if you put a premium on managing server resources, or if the number or size of messages might exhaust available memory.

You can also control memory use by reducing the value of **Batch Size** on the receive location. A smaller batch size keeps fewer messages in memory and therefore uses less memory.

Placing send ports and receive locations on separate computers can also reduce memory use.

Send Optimization

On the send side, you can achieve the equivalent single-message processing by using the sample orchestration. The sample sends a single message and then waits to send the next message until it receives an acknowledgment.

Sending and Receiving Large Messages

The MSMQ adapter default message handling depends, in part, on the size of the message. When a message is less than four megabytes (4 MB), the MSMQ adapter uses the .NET Framework Class Library. Otherwise, it uses the large message extensions in Microsoft BizTalk Server.

If your application consistently receives or sends large messages, you may have to control the amount of memory that the adapter uses. For more information about conserving memory,

Setting Up the MSMQT and MSMQ Adapters on One Computer

You can use the MSMQ adapter and the Microsoft BizTalk Message Queuing adapter (MSMQT) on the same computer. Because both adapters use the same port, you have to add an IP address for the adapter for MSMQ to use.

To set up a computer for MSMQ and MSMQT

1. Add one additional static IP address to the computer.
2. Create an entry in the Domain Name System (DNS) for the new IP address. Make sure that you assign a name, such as MSMQServer, to clearly identify it.
3. Install Message Queuing on the computer.
4. Use Registry Editor to create a new registry value called **BindInterfaceIP** under [**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters**], and assign the IP address that you just created to the new registry value.
5. Reconfigure BizTalk Server: Use the computer name as the DNS name assigned to MSMQT and use its corresponding IP address.
6. If you want to use both the MSMQ adapter and MSMQT, you will need to add MSMQT back to the adapter list in the BizTalk Server Administration console.

Reliable Messaging with the MSMQ Adapter

You can improve the reliability of sending and receiving messages with the MSMQ adapter by using particular configuration settings and by using transactions.

In This Section

- Properties for Reliable Messaging with the MSMQ Adapter
- Transaction Handling with the MSMQ Adapter

Properties for Reliable Messaging with the MSMQ Adapter

You can improve the reliability of sending and receiving messages with the MSMQ adapter by the way you configure the MSMQ adapter. This topic discusses using several configuration properties for reliable messaging.

Running MSMQ Adapter Handlers Within a Clustered BizTalk Host

One approach to high availability is to run adapter handlers in multiple host instances on different BizTalk servers simultaneously. This approach is not recommended for the MSMQ adapter handlers, because MSMQ does not support remote transacted reads and because the MSMQ send handler maintains a dependency on the locally running instance of the MSMQ service. To provide high availability for the MSMQ send and receive handlers it is recommended that you run the MSMQ adapter handlers in a clustered instance of a BizTalk Host. For more information,

Queue Failure and the Dead Letter Queue

After successfully sending a message, there is no error for subsequent messages if the receiving queue is disabled or deleted. This situation could cause loss of messages.

Setting the **Use Dead Letter Queue** configuration property to **True** prevents you from losing messages. When the property is **True** (the default), messages that the queue does not receive go into the dead letter queue.

Impersonation and Remote Queues

You also have to set the **Use Dead Letter Queue** configuration property to **True** when you use remote queues. If the adapter for MSMQ impersonates a user without permission to use the remote queue, the message could be lost.

When the property is **True** and the impersonated user does not have permission to use the remote queue, the message goes to the dead letter queue on either the local or remote computer. In a transactional send, the message goes to the dead letter queue on the local computer. In a non-transactional send, the message goes to the dead letter queue on the remote computer.

Recoverable and Use Journal Queue Properties

Both the **Recoverable** and **Use Journal Queue** properties save copies of sent messages. For more information about these properties,

Transaction Handling with the MSMQ Adapter

This section discusses how transactions work in receiving and sending.

You can use transactions on both send and receive with the MSMQ adapter. On transacted sends, the adapter accumulates messages until it has a complete batch. The adapter then submits the batch to the local Message Queuing service as a single transaction. If the submission fails, the adapter tries to resubmit the batch. If the resubmission fails, the adapter moves to the secondary transport.

On transacted receives, the adapter suspends failed messages so that it does not lose any one of the messages. During a transacted receive the adapter adds messages to a batch until the batch is complete. It then submits the batch:

- If there are no problems and the server receives the messages, the adapter commits the transaction.
- If there are problems, the adapter creates a new batch as part of the current transaction. It then moves any problem messages into the new batch. It then resubmits the first batch and submits the new batch as suspended messages. The adapter commits the transaction if there are no problems during this resubmission.

If you are running an MSMQ adapter send handler in a clustered BizTalk Host instance, you should cluster the MSMQ service in the same cluster group to ensure transactional consistency.

If you are running an MSMQ adapter receive handler in a clustered BizTalk Host instance, you should cluster the MSMQ service in the same cluster group to support local transacted reads because MSMQ does not support remote transactional reads. For more information about running MSMQ adapter handlers in a clustered instance of a BizTalk Host,

Analyzing MSMQ Adapter Errors with the Trace Tool

You use the trace tool to analyze messaging failures when you run your application. You use trace.cmd for the MSMQ adapter and your BizTalk Server application. The tool uses tracelog.exe. You have to install tracelog.exe if you do not already have it.

Install the Trace Utility

To install the BizTalk Adapter Trace Utility, follow these steps:

1. To download the Tracelog.exe file, visit the following Microsoft Platform SDK download Web site: http://go.microsoft.com/fwlink/?LinkId=21975.
2. Start the Platform SDK Web installation program by clicking the link for the **PSDK-x86.exe** file at the bottom of the Web page.
3. When you are prompted, choose the option for a custom installation.

4. In the **Custom Installation** dialog box, click to clear all the available features.

5. Expand the **Microsoft Windows Core SDK** feature, and then expand the **Tools** feature.

6. Choose the **Tools (Intel 64-bit)** feature, and then click **Will be installed on local hard drive**.

7. Click **Next**, and then click **Next** again to start the installation.

8. Locate the *Drive***:\MicrosoftPlatformSDKInstallationFolder\bin** folder, and then copy the Tracelog.exe file to the Microsoft BizTalk Server installation folder. The BizTalk Server installation folder also contains the Trace.cmd file.

Enable the Trace Utility

To enable the BizTalk Adapter Trace Utility in BizTalk Server, follow these steps:

1. At a command prompt, change the current directory to the directory where BizTalk Server is installed. By default, BizTalk Server is installed in the Program Files\Microsoft BizTalk Server directory.

2. Type the following command, and then press ENTER:

   trace -tools "*Path of the BizTalk Adapter Trace Utility***"**

   By default, the BizTalk Adapter Trace Utility is located in the C:\Program Files\Microsoft Platform SDK\Bin directory. You must enclose the path of the BizTalk Adapter Trace Utility in quotation marks.

   For example, type the following command:

   trace -tools "C:\Program Files\Microsoft Platform SDK\Bin"

   The **-tools** switch indicates to the Trace.cmd file the location of the Tracelog.exe file.

Run the Trace Utility

To run the BizTalk Adapter Trace Utility, follow these steps:

1. At a command prompt, type the following command, and then press ENTER:

   trace -start

2. Reproduce the scenario that you want to trace.

3. At a command prompt, type the following command, and then press ENTER:

   trace -stop

After you stop the trace, a binary file that is named Bts2006.bin is generated in the folder where BizTalk Server is installed.

4. The bts2006.bin file contains the output of the trace tool. Contact Microsoft Product Support Services for analysis.

MSMQ Adapter Property Schema and Properties

The MSMQ adapter assigns values to context properties that you use in your applications. For a list of the send and receive properties in the MSMQ adapter.

Context Properties

The following table shows the context properties to which the MSMQ adapter assigns values.

Name	Type	Description	Promoted
Acknowledgement	int	Specifies the classification of acknowledgment that this message represents using the values in the **System.Messaging.Acknowledgment** enumeration	No
AcknowledgeType	int	Specifies the type of acknowledgment message that the sending application requests.	No
AdministrationQueue	string	Specifies the name of the queue name that receives the acknowledgment message.	No
AppSpecific	int	Specifies application-specific information that you can use to organize different types of messages.	Yes
ArrivedTime	dateTime	Specifies the time that the message arrived in the destination queue.	No
Authenticated	boolean	Specifies whether the message was authenticated.	No
BodyType	Int	Specifies the type of data that the message body contains.	No
CertificateThumbPrint	string	Specifies the thumbprint of the client certificate that you want to use for message authentication purposes.	Yes

CorrelationId	string	Specifies the message identifier used by acknowledgment, report, and response messages to reference the original message.	Yes
EncryptionAlgorithm	int	Specifies the encryption algorithm used to encrypt the body of a message.	No
Id	string	Specifies the message's identifier.	No
Label	string	Specifies an application-defined Unicode string that describes the message.	Yes
MaximumMessageSize	unsignedint	Specifies the maximum message size in kilobytes for messages that you send to the specified queue.	No
MessageType	int	Specifies the message type. A Message Queuing message can be one of the following types: • Normal, which is either a typical message sent from an application to a queue, or a response message returned to the sending application. • Acknowledgement, which Message Queuing generates whenever the sending application requests one. For example, Message Queuing can generate positive or negative messages to indicate that the original message arrived or was read. Message Queuing returns the appropriate acknowledgment message to the administration queue specified by the sending application. • Report, which Message Queuing generates whenever a report queue is defined at the source Queue Manager. When tracing is enabled, Message Queuing sends a report message to the Message Queuing report queue each time the original message enters or leaves a Message Queuing server.	No
Priority	int	Specifies the message priority using the values defined in the **System.Messaging.MessagePriority** enumeration.	Yes

Recoverable	boolean	Specifies whether the message is guaranteed to be delivered in the event of a computer failure or network problem.	No
ResponseQueue	string	Specifies the queue that receives application-generated response messages.	No
SegmentationSupport	boolean	Specifies whether the segmentation of messages larger than 4 MB is supported.	No
SentTime	dateTime	Specifies the date and time on the sending computer that the message was sent by the source queue manager.	No
SourceMachine	string	Specifies the computer from which the message originated.	No
TimeOut	int	Specifies the time for the message to reach the destination queue before a time-out occurs.	No
TimeOutUnits	string	Specifies the units for the **TimeOut** property. You can set the property to Days, Hours, Minutes, or Seconds.	No
Transactional	boolean	Specifies the behavior for transactional and non-transactional send ports and receive locations.	No
UseAuthentication	boolean	Specifies whether the message was (or must be) authenticated before being sent.	No
UseDeadLetterQueue	boolean	Specifies whether a copy of the message that could not be delivered should be sent to a dead-letter queue.	No
UseJournalQueue	boolean	Specifies whether a copy of the message should be kept in a machine journal on the originating computer.	No

Message Labels

You can use the Message Queuing **Label** property in filters by adding a reference to MSMQAdapterProperties.dll and selecting the property in the **Filter** dialog box. You can also use the property in other contexts because the MSMQ adapter automatically adds it to the message context.

Ordered Delivery of Messages with the MSMQ Adapter

BizTalk Server 2006 provides an **Ordered Delivery** option for static send ports. Setting the **Ordered Delivery** option on a send port to **True** ensures that BizTalk Server delivers messages to the send port in the same order that they are published to the MessageBox database. To provide end-to-end ordered delivery the following conditions must be met:

- Messages must be received with an adapter that preserves the order of the messages when submitting them to BizTalk Server.
- You must subscribe to these messages with a send port that has the **Ordered Delivery** option to **True**.
- If an orchestration is used to process the messages, only a single instance of the orchestration should be used, the orchestration should be configured to use a sequential convoy, and the **Ordered Delivery** property of the orchestration's receive port should be set to **True**.

Using the MSMQ Adapter for Ordered Delivery of Messages

The MSMQ receive adapter provides support for preserving the order of messages when submitting them to BizTalk Server. End-to-end ordered delivery of messages through BizTalk Server can be achieved when receiving messages with the MSMQ adapter if the messages are processed by a send port that is configured with the **Ordered Delivery** option set to **True**.

POP3 Adapter

You use the Post Office Protocol 3 (POP3) adapter to retrieve data from a server that houses POP3 mailboxes into a server running Microsoft BizTalk Server by means of the POP3 protocol.

The POP3 adapter consists of only one adapter, a receive adapter. The receive adapter controls the receive locations that use the POP3 adapter.

This topic discusses the workflow of the POP3 receive adapter.

POP3 Receive Adapter

The POP3 receive adapter retrieves e-mail from a specified mailbox on a specified POP3 server. By default, the POP3 receive adapter applies MIME processing to the e-mail messages that it downloads and submits these messages to BizTalk Server as multipart BizTalk messages. The POP3 receive adapter can receive and process e-mail in the following formats:

- Plain text
- MIME encoded
- MIME encrypted

- MIME encoded and signed
- MIME encrypted and signed

Batching Support for the POP3 Receive Adapter

The POP3 receive adapter does not support batching.

Authentication with POP3 Server

The following authentication methods are supported for use with the POP3 adapter:

- **Basic.** The POP3 server uses user provided credentials for authentication. These credentials are sent in clear text.
- **Digest (APOP).** The POP3 server uses a digest string for authentication.
- **Secure Password Authentication (SPA).** The POP3 server uses current process credentials for authentication.

In This Section

- What Is the POP3 Adapter?
- Configuring the POP3 Adapter
- Walkthrough: Creating a BizTalk Application That Uses the POP3 Adapter

What Is the POP3 Adapter?

This section describes the POP3 receive adapter.

POP3 Receive Adapter

The POP3 receive adapter enables you to move data from a POP3-enabled mailbox to BizTalk Server.

The key features of the POP3 receive adapter are:

- Pulling files from the POP3 server mailbox on demand.
- Running polls based on a configurable schedule.
- Polling the POP3 server mailbox and sending data directly to BizTalk Server.
- Specifying the POP3 server mailbox as an IP address or host name, port, user name, and password.

- Ability to download e-mail from mail servers that require Secure Sockets Layer (SSL) connections.
- Guaranteed file delivery.
- Implicit MIME processing. It is not necessary to use a MIME decoder in a receive pipeline when using the POP3 adapter.

POP3 Adapter Supported Platforms

The POP3 adapter is designed to work with any POP3 servers that conform to the following RFCs:

- **RFC 1939.** Post Office Protocol Version 3
- **RFC 1734.** POP3 AUTHentication command
- **RFC 2045.** Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
- **RFC 2046.** Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
- **RFC 2047.** MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text

The POP3 adapter was tested extensively against Microsoft Exchange Server 2003.

Body Part Selection Algorithm Used by the POP3 Adapter

The POP3 adapter creates a multipart BizTalk message from the parts of the MIME-encoded message that it receives. When the POP3 adapter creates the multipart BizTalk message, it selects one of the message parts as the BizTalk message body part.

The POP3 adapter selects the BizTalk message body part from the available body parts based upon the values supplied for the **Body Part Index** and the **Body Part Content Type**.

The algorithm that is used to select the BizTalk message body part of an e-mail is described below:

1. If the **Body Part Index** is set to 0 and the **Body Part Content Type** is blank then the following algorithm is used to select the BizTalk message body part:
   - Use the first MIME part with the Content-Description header set to "body".
   - Otherwise use the first MIME part with the Content-Type header set to "text/xml".
   - Otherwise use the first MIME part with the Content-Type header set to "text/plain".

- Otherwise use the first MIME part with the Content-Type header set to "text/".
- Otherwise use the first MIME part.

2. Otherwise if the **Body Part Index** is set to 0 and the **Body Part Content Type** is set, then the first body part of the incoming message that matches the specified **Body Part Content Type** is selected as the BizTalk message body part. If there are no parts with a matching content type then the message is suspended.

3. Otherwise if the **Body Part Index** is set to a value greater than 0 and the **Body Part Content Type** is blank, then the body part with the specified index is selected as the BizTalk message body part. If the specified index is greater than the number of body parts then the message is suspended.

4. Otherwise if the **Body Part Index** is set to a value greater than 0 and the **Body Part Content Type** is set, then the **Body Part Index** is only applied to those body parts that match the specified **Body Part Content Type** and the corresponding body part is selected as the BizTalk message body part. If the specified index is greater than the number of parts with a matching content type then the message is suspended. If there are no parts with a matching content type then the message is suspended.

Considerations for Preventing Data Duplication When Using the POP3 Adapter

The POP3 adapter is not a transactional adapter and therefore is subject to processing multiple copies of the same message, potentially causing data duplication. It is possible that the POP3 adapter will deliver duplicate copies of a message in the following scenarios:

- The POP3 adapter always deletes e-mails from the mailbox that it is configured to monitor after the e-mail has been successfully submitted to BizTalk Server for processing. If the POP3 adapter retrieves an e-mail from a mailbox, submits the e-mail to BizTalk Server for processing, and fails to delete the e-mail from the mailbox, the e-mail will be resubmitted to BizTalk Server the next time that the POP3 adapter polls the mailbox.

- If multiple instances of the POP3 adapter in separate BizTalk Host instances are monitoring the same mailbox simultaneously and the POP3 server allows multiple concurrent connections to its mailboxes, then the adapter may deliver duplicate copies of messages.

High Availability for the POP3 Adapter

Some POP3 servers permit multiple concurrent connections to a given mailbox. If more than one instance of the POP3 adapter is configured to retrieve mail from a mailbox on such a POP3 server then data duplication can occur. Therefore you should configure only one instance of the POP3 adapter to retrieve mail from a mailbox that permits multiple concurrent connections.

To provide fault tolerance for the POP3 adapter in this scenario, a single POP3 adapter receive handler should be configured to run in a clustered BizTalk Host.

Authentication Warnings When Multiple Instances of the POP3 Adapter Connect to the Same Mailbox

BizTalk Server may be configured to have more than one instance of the POP3 adapter retrieve mail from the same mailbox. In such a case, it is possible that authentication warnings may be generated in the BizTalk Server Application log because of the locking mechanism employed by some POP3 servers. These warnings will have no impact on the POP3 adapter functionality and can be safely ignored in this scenario.

Configuring the POP3 Adapter

This section describes how to configure a POP3 adapter.

In This Section

This section contains:

- How to Configure a POP3 Receive Handler
- How to Configure a POP3 Receive Location
- POP3 Adapter Property Schema and Properties

How to Configure a POP3 Receive Handler

Use the following procedure to change the host associated with the POP3 receive handler.

To configure the general properties for a POP3 receive handler

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.
2. In the expanded adapter list, click **POP3**, in the right pane, right-click the receive handler that you want to configure, and then click **Properties**.
3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.
4. Click **OK**.

How to Configure a POP3 Receive Location

You can set POP3 receive location adapter variables in the BizTalk Server Administration console. If properties are not set in the receive location, the default receive handler values set in the BizTalk Server Administration console are used.

To configure variables for a POP3 receive location

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application you want to create a receive location in.

2. In the BizTalk Server Administration console, in the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

3. In the **Receive Port Properties** dialog box, in the left pane, select **Receive Locations**, and then in the right pane, double-click an existing receive location or click **New** to create a new receive location.

4. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **POP3** from the drop-down list, and then click **Configure**.

5. In the **POP3 Transport Properties** dialog box, do the following:

Use this	To do this
Apply MIME Decoding	Specify whether to apply MIME decoding to messages received by the POP3 adapter. MIME decoding is used to parse the incoming message and any attachments into a multipart BizTalk message. Default value: **True**
Body Part Content Type	Specify the body part content type of the incoming e-mail message to submit to BizTalk Server. This is an optional setting.
Body Part Index	Specify the body part of the incoming e-mail message to submit to BizTalk Server. Default value: 0
Mail Server	Specify the POP3 mail server that houses the mailbox that will be polled by the POP3 adapter.
Port	Specify the port for the POP3 mail server. Valid values: 1 through 65535 inclusive. Default value: 0
Authentication Scheme	Specify the type of authentication to use with the destination server.

	Valid options are: • **Basic** • **Digest** • **SPA**
Password	Specify the user password to use for authentication with the POP3 server.
Use SSL	Specify whether to use Secure Sockets Layer (SSL) to communicate with the destination server. Default value: **False**
User Name	Specify the user name to use for authentication with the POP3 server. This property requires a value.
Error Threshold	Specify the maximum number of network or protocol errors to wait before shutting down the adapter. Specify a value of 0 to prevent the adapter from shutting down. Default value: 10
Polling Interval	Specify the interval between attempts to retrieve messages from the POP3 server. Default value: 5
Polling Interval Unit	Specify the unit of measure to be used for the **Polling Interval**. Default value: Minutes

6. Click **OK**.

7. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and then click **OK** to save settings. For information about the **Receive Locations Properties** dialog box,

POP3 Adapter Property Schema and Properties

The following table lists the properties in the POP3 adapter property schema.

Namespace: http://schemas.microsoft.com/BizTalk/2003/pop3-properties

Name	Type	Description
Subject	xs:string	Specifies the content placed on the **Subject** header for the message
From	xs:string	Specifies the e-mail address placed on the **From** header field of the e-mail message.
To	xs:string	Specifies the e-mail address or addresses placed on the **To** header field of the e-mail message.
ReplyTo	xs:string	Specifies the e-mail address placed on the **ReplyTo** header field of the e-mail message.
CC	xs:string	Specifies the e-mail address or addresses placed on the **CC** header field of the e-mail message.
Date	xs:string	Specifies the content placed on the **Date** header field of the e-mail message.
DispositionNotificationTo	xs:string	Specifies the content placed on the **DispositionNotificationTo** header field of the e-mail message.
Headers	xs:string	Specifies the content of all of the header fields of the e-mail message.

Walkthrough: Creating a BizTalk Application That Uses the POP3 Adapter

This section takes you through creating a simple Microsoft BizTalk Server 2006 application using the POP3 adapter.

This application assumes that you have not yet created any send ports or receive locations. If you have existing send ports or receive locations, substitute appropriate names when you work through the steps.

The application is a simple content-based routing application using only a receive location and a send port. The receive location reads from a mailbox on the server running Windows Server 2003 ("the Windows server"). The send port takes the message from the receive location and sends it to a folder on the local file system of the BizTalk server.

To create the application, you have to create the mailbox, set up the BizTalk Server receive location and send port, start the send port and enable the receive location, and send a test message to the mailbox.

To create a mailbox on Windows Server 2003 with Email Services installed, perform the following procedure.

Create a mailbox on Windows Server 2003

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **POP3 Service**.

2. Expand <*servername*> and click on the domain where you would like to create a mailbox.

3. In the **POP3 Service** dialog box, in the right pane, click the **Add Mailbox** option.

4. In the **Add Mailbox** dialog box, in the **Mailbox Name** box, type **EmailTest**.

5. Select the **Create associated user for this mailbox** check box.

6. In the **Password** and **Confirm Password** boxes, type a password, and then click **OK**.

7. Make a note of the **Account name** and **Mail Server** log on information displayed for use with clear text authentication in the **POP3 Service** dialog box, and then click **OK**. This information will be used by the BizTalk Server receive location that you configure with the POP3 transport type.

The next steps create the receive location and the send port, and start the send port and enable the receive location.

Create the receive location

1. Click **Start**, point to **Programs**, point to **Microsoft Visual Studio 2005**, and then click **Microsoft Visual Studio 2005**.

2. On the **View** menu, click **BizTalk Explorer**.

3. Double-click the default database. The default database name has the form **<***machine_name***>.BizTalkMgmtDb.dbo** where *machine_name* is the name of your computer.

4. Right-click **Receive Ports**, and then click **Add Receive Port**.

5. In the **Create New Receive Port** dialog box, in the **Specify the type of Receive Port** box, select **One-Way Port**, and then click **OK**.

6. In the **One-Way Receive Port Properties** dialog box, in the **Name** box, type **POP3Receive**, and then click **OK**.

7. Right-click **Receive Locations**, and then click **Add Receive Location**.

8. In the **Receive Location Properties** dialog box, in the **Name** box, type **POP3Receive**.

9. In the **Transport Type** box, select **POP3**.

10. In the **Receive Handler** box, select **BizTalkServerApplication**.

11. In the **Receive Pipeline** box, select **Microsoft.BizTalk.DefaultPipelines.PassThruReceive**.

12. In the **Address (URI)** box, click the ellipsis (**...**) button.

13. In the **POP3 Transport Properties** dialog box, in the **Apply MIME Decoding** box, select **False**.

14. In the **Mail Server** box, type the name of the Windows Server 2003-based server where you created a mailbox.

15. In the **Authentication Scheme** box, select **Basic**.

16. In the **Password** box, click the drop-down arrow and type the password for the mailbox.

17. In the **User Name** box, type the fully qualified user name for the mailbox, for example *username@host.domain.toplevel_domain.*

18. In the **Polling Interval** box, type **1**, click **OK**, and then click **OK** again.

Create the send port and destination folder on the BizTalk server

1. Create a folder on the BizTalk Server file system. This will be the destination for the send port.

2. Right-click **Send Ports**, and then click **Add Send Port**.

3. In the **Create New Send Port** dialog box, in the **Specify the type of Send Port** box, select **Static One-Way Port**, and then click **OK**.

4. In the **Static One-Way Send Port Properties** dialog box, in the **Transport Type** box, select **FILE**.

5. In the **Name** box, type **SendToFile**.

6. In the **Address (URI)** box, click the ellipsis (**...**) button.

7. Next to the **Destination folder** box, click **Browse**, select the folder that you created on the BizTalk server, and then click **OK**.

8. In the **File name** box, type **%MessageID%.txt**, and then click **OK**.

9. Click **Send**, and in the **Send Pipeline** box, select **Microsoft.BizTalk.DefaultPipelines.PassThruTransmit**.

10. Click **Filters & Maps**, and then click **Filters**.

11. In the **Property** box, select **BTS.ReceivePortName**.

12. In the **Value** box, type **POP3Receive**, and then click **OK**.

Enable the receive location and start the send port

1. Right-click the **POP3Receive** receive location, and then click **Enable**.

2. Right-click the **SendToFile** send port, and then click **Start**.

The next step is to test the application by sending a test message to the mailbox monitored by the receive location.

Configure Outlook Express to send an e-mail message to the mailbox

1. Click **Start**, point to **Programs**, and then click **Outlook Express**.

2. In Outlook Express, on the **Tools** menu, click **Accounts**.

3. Click **Add** and then click **Mail**.

4. In the **Display name** box, type a display name, and then click **Next**.

5. In the **Internet E-mail address** dialog box, in the **E-mail address** box, type **EmailTest@<domain_name>**, and then click **Next**.

   Make sure to enter the appropriate value for <*domain_name*>. This value should match the name of the domain under which this mailbox was created in the POP3 Service Administration interface on the Windows server.

6. In the **E-mail Server names** dialog box, in the **Incoming mail** and **Outgoing mail** boxes, type the server name or IP address of the Windows server, and then click **Next**.

7. In the **Internet Mail Logon** dialog box, in the **Account name** box, type **EmailTest**.

8. In the **Password** box, type the password for the EmailTest account, select the **Remember password** option, click **Next**, and then click **Finish**.

9. Click to select the account that you just created, and then click **Properties**.

10. In the **Properties** dialog box, click the **Advanced** tab, click to select the option to **Leave a copy of messages on the server**, and then click **OK**.

11. In the **Internet Accounts** dialog box, click **Close**.

12. Use Outlook Express to compose a test message, type **Test** into the **Subject** field, and type **EmailTest@<domain_name>** into the **To** field.

13. Click **Send** to send the test message. To ensure that Outlook Express sends the test message immediately, click the **Send/Recv** button in the Outlook Express toolbar.

View the message

1. Use Windows Explorer to open the folder that you specified as the **Destination Folder** for the send port.

2. Double click the document in the folder to view the contents of the document in Notepad.

SMTP Adapter

You use the Simple Mail Transfer Protocol (SMTP) adapter to exchange information between a server running Microsoft BizTalk Server and other applications by means of the SMTP protocol. BizTalk Server can send messages to other applications by creating an e-mail message and delivering it to a specified e-mail address. Internally, the SMTP send adapter creates an SMTP-based e-mail message and sends it to a target e-mail address. The target e-mail address is a property of the SMTP adapter. BizTalk Explorer exposes this property when you configure the SMTP send port.

The SMTP adapter supports wildcard characters in the **TO**, **FROM**, **CC** and **SUBJECT** properties, and resolves them to their actual values. If wildcard characters in the **TO**, **FROM**, and **CC** properties cannot be resolved, the SMTP transport logs an error and puts the message into the suspended queue or redirects the message to the backup transport. If the wildcard characters cannot be resolved in the **SUBJECT** property, the message is sent with the **SUBJECT** property specified exactly as in the property (for example, "Message %MessageID%").

By default, the message text of SMTP messages is plain text. To use HTML in message bodies, you can configure the adapter to use the contents of an HTML file for the message text.

The SMTP adapter consists of only one adapter, a send adapter. The send adapter controls the send ports that use the SMTP adapter.

This topic discusses the flow of a message through the SMTP send adapter.

SMTP Send Adapter

The SMTP send adapter gets messages from the server and posts them to an SMTP server that sends them to e-mail recipients. The SMTP send adapter gets the message content from the body part of the BizTalk Message object, from a specified file, or from a text entered into a dialog box that is available when configuring the adapter.

After the SMTP send adapter successfully posts a message onto the SMTP server, the SMTP send adapter deletes the message from the MessageBox database.

The SMTP send adapter can request delivery notification and read receipts for messages sent over the SMTP send adapter. The SMTP adapter delivers the notification and read receipt to the address specified on the SMTP **From** header.

Authentication with SMTP Server

If you require SMTP server authentication, the SMTP send adapter uses one of the following authentication types:

- **Basic.** The SMTP server uses user-provided credentials for authentication.
- **Process account (NTLM).** The SMTP server uses current process credentials for authentication.

In This Section

- Configuring the SMTP Adapter
- Restrictions When Configuring the SMTP Adapter
- SMTP Adapter Security Recommendations

Configuring the SMTP Adapter

This section describes how to configure an SMTP adapter.

In This Section

- How to Configure an SMTP Send Handler
- **Configuring an SMTP Send Port**
- SMTP Adapter Property Schema and Properties

How to Configure an SMTP Send Handler

You can set SMTP send handler properties in the BizTalk Administration console. These send handler properties are used as the send port configuration values if the properties are not set on the individual SMTP send port.

To change global variables for an SMTP send handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.
2. In the expanded adapter list, click **SMTP**, in the right pane, right-click the send handler that you want to configure, and then click **Properties**.
3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the send handler will be associated, and then click **Properties**.

4. In the **SMTP Transport Properties** dialog box, on the **Properties** tab, do the following:

Use this	To do this
SMTP server name	Specify the name of the SMTP server to use when sending messages. This property requires a value. Maximum length: 256
From (e-mail address)	Required. Specify the e-mail address to place on the SMTP **From** header. Maximum length: 256
Authentication type	Specify the type of authentication to use with the SMTP server. Options: • **No authentication** • **Basic authentication** • **Process account (NTLM)** Default value: Process account (NTLM)
User name	Specify the user name to use for authentication with the SMTP server. This property requires a value if **Authentication type** is **Basic authentication**. Minimum length: 0 Maximum length: 256
Password	Specify the password to use for authentication with the SMTP server. This property requires a value if **Authentication type** is **Basic authentication**. Minimum length: 0 Maximum length: 256

5. Click **OK**.

How to Configure an SMTP Send Port

You can configure an SMTP send port either programmatically or by using the BizTalk Server Administration console.

How to Configure an SMTP Send Port Programmatically

The SMTP adapter stores its configuration information in the BizTalk Management database (also known as the Configuration database). Configuration information is stored in a custom XML property bag. During initialization of the SMTP adapter and during its run time, the server passes the configuration to the adapter as follows:

- For the SMTP send handler, configuration information passes to the adapter by calling the **Load** method of the **IPersistPropertyBag** interface.

- For the SMTP send adapters, configuration information passes to the adapter as a set of properties on a message context. The SMTP namespace groups these properties together.

The BizTalk Explorer object model exposes the **ITransportInfo** adapter configuration interface for send ports, which contains the **TransportTypeData** read/write property. This property accepts the SMTP send port configuration property bag as a name/value pair XML string. Note that to set this property in the BizTalk Explorer object model, it must first be set on the **Address** property of the **ITransportInfo** interface.

Setting the **TransportTypeData** property of the **ITransportInfo** interface is not required. If it is not set, the SMTP send port uses the default values for the SMTP send handler. SMTP send port-specific properties are defined in SMTP send adapter property schema bts_smtp_properties.xsd.

If you do not define properties that duplicate the send handler configuration properties, configuration properties for the handler are used. If you do not define required properties, default values are used. If you do not define default values, the SMTP send handler logs an error in the event log and moves the message to the backup adapter.

You can set these properties programmatically on a message context. You can set these properties in a BizTalk orchestration schedule or in a custom pipeline component. The following rules apply when using these properties:

- If the property is set in an orchestration or in a custom pipeline component in a receive pipeline, then:

  - If the message is sent to a static send port, the property value will be overwritten with the value configured for that send port.

  - If the message is sent to a dynamic send port, the property value will not be overwritten.

- If the property is set in a custom pipeline component in a send pipeline, then:
  - The value will not be overwritten regardless of whether the message is sent to a static or dynamic send port.

The following table lists the configuration properties that you can set in the BizTalk Explorer object model for the SMTP send location.

Property name	Type	Description	Restrictions	Comments
SMTPHost	xs:string	SMTP server used to send messages.	Maximum length: 256	Default value: Empty. The default value indicates t use the configuration values
From	xs:string	The e-mail address that the SMTP send port places on the SMTP **From** header.	Maximum length: 256	Default value: Empty. The default value indicates t use the configuration values
CC	xs:string	E-mail address where a copy of the message will be sent.	Maximum length: 1024	Default value: Empty You can list several e-mail a
Subject	xs:string	Subject header for the messages.	Minimum length: 0 Maximum length: 256	Default value: %MessageID%
SMTPAuthenticate	xs:int	Type of authentication to use.	None	Valid values: • 0 - No authentication • 1- Basic authenticatio • 2 - Process account (The default value indicates t use the configuration values the default value, omit this bag when setting the **Transp**
UserName	xs:string	User name to use for authentication with the SMTP server.	Minimum length: 0 Maximum length: 256	Default value: Empty Requires a value if **SMTPA** (Basic authentication).

Password	xs:string	User password for authentication with the SMTP server.	Minimum length: 0 Maximum length: 256	Default value: Empty Requires a value if **SMTPA** (Basic authentication).
ReadReceipt	xs:boolean	Requests a read receipt for the messages from this send port.	None	Default value: False
DeliveryReceipt	xs:boolean	Requests a delivery receipt for the messages from this send port.	None	Default value: False
EmailBodyText	xs:string	Specify text to be used for the body of the e-mail being sent.	Maximum length: 64 kb	Default value: Empty
EmailBodyTextCharset	xs:string	Specify the character set to use for encoding the body of the e-mail being sent when the **EmailBodyText** option is used. The SMTP adapter will convert the **EmailBodyText** to the character set specified by **EmailBodyTextCharset**.	None	Default value: UTF-8 (65001
EmailBodyFile	xs:string	Specifies that the contents of a file will be used for the body of the e-mail being sent and the full path to the file. This path must be accessible to the host for the SMTP adapter at run time.	Maximum path length: 256 characters	Default value: Empty
EmailBodyFileCharset	xs:string	Specify the character set to use for encoding the body of the e-mail being sent if the **EmailBodyFile** property is set. The SMTP adapter will not perform any conversion on the file; the file must already be encoded in this character set. If the file has a Byte-Order-Mark (BOM), the SMTP adapter will remove	None	Default value: UTF-8 (65001

		it.		
Attachments	xs:string	Specifies that a file or files will be attached to the e-mail message and the full path to the file or files. The specified path or paths must be accessible to the host for the SMTP adapter at run time.	Maximum path length: 256 characters	Default value: Empty
MessagePartsAttachments	xs:int	Specify how BizTalk message parts are attached to the e-mail message	None	Valid values: • 0 - No BizTalk mess attachments. • 1- The BizTalk messa e-mail attachment. In th or **EmailBodyText** prope neither of these properti message body part is instead of as an attachme • 2 - All parts are sen if **EmailBodyText** or specified, then the BizT sent as the e-mail body attachments. Default value: 0
ReplyBy	xs:dateTime	Populates the **Reply-By** header field in the outgoing message with the specified value.	This property cannot be set on the send port property page. This property can be set from a pipeline or an orchestration.	Default value: Empty

The following code shows the format of the XML string to use to set these properties:

How to Configure an SMTP Send Port with the BizTalk Server Administration Console

You can set SMTP send port adapter variables in the BizTalk Server Administration Console. If properties are not set for the send port, the default send handler values set in the BizTalk Server Administration Console are used.

To configure an SMTP send port with the BizTalk Server Administration console, use the following procedure.

To configure variables for an SMTP send port

1. In the BizTalk Server Administration Console, create a new send port or double-click an existing send port to modify it. for more information. Configure all of the send port options and specify **SMTP** for the **Type** option in the **Transport** section of the **General** tab.

2. On the **General** tab, in the **Transport** section, next to **Type**, click **Configure**.

3. In the **SMTP Transport Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
To	Required. Specify the e-mail address for where to send messages. You can specify more than one address. Maximum length: 256 .
CC	Specify the e-mail address to send the carbon copy of the message. You can specify more than one address. Maximum length: 1024
Subject	Specify the subject header for the message. Minimum length: 0 Maximum length: 256
Notification	Specify the type of notification receipt. You can select one or both types of receipts. Notification receipt types are: • **Read Receipt**. Confirmation e-mail message is sent when the message is read. • **Delivery Receipt**. Confirmation e-mail message is sent when the message is delivered.

4. In the **SMTP Transport Properties** dialog box, on the **Compose** tab, do the following:

Use this	To do this
BizTalk message body part	Specify to use the BizTalk message body part for the body of the e-mail being sent.
Text	Specify text to be used for the body of the e-mail being sent. After the **Text** option is selected you can enter the text for the e-mail body into the text box. **Maximum Length:** 64Kb
Charset for the text	• Specify the character set to use for encoding the body of the e-mail being sent. This option is only available if the **Text** option is selected. • **Default value:** UTF-8 (65001)
File	Specify that the contents of a file will be used for the body of the e-mail being sent and specify the path to the file. After the **File** option is selected you can click the Ellipsis (**...**) button to browse to the file. Maximum path length: 256 characters
Charset of the file	Specify the character set encoding of the file being sent. This option is only available if the **File** option is selected. Default value: UTF-8 (65001)

5. In the **SMTP Transport Properties** dialog box, on the **Attachments** tab, do the following:

Use this	To do this
Remaining BizTalk message parts	Specify how BizTalk message parts are attached to the e-mail message. Options: • **Do not attach parts** • **Attach only body part** • **Attach all parts** Default value: Do not attach parts.

Add	Specify a file or files to attach to the e-mail message. After clicking **Add** you can browse to select a file and add it to the list of files to be attached. Maximum path length: 256 characters
Remove	Removes the selected file from the list of files to be attached to the e-mail message.

6. In the **SMTP Transport Properties** dialog box, on the **Handler Override** tab, do the following:

Use this	To do this
SMTP server name	Specify the name of the SMTP server to use when sending messages. Maximum length: 256
From (e-mail address)	Specify the e-mail address to place on the SMTP **From** header. Maximum length: 256
Authentication type	Specify the type of authentication to use with the SMTP server. Options: • **(Default)** • **No authentication** • **Basic authentication** • **Process account (NTLM)** The default value indicates that the SMTP send port will use the configuration values specified in the send handler.
User name	Specify the user name to use for authentication with the SMTP server. This property requires a value if **Authentication type** is **Basic authentication**.

	Minimum length: 0 Maximum length: 256
Password	Specify the password to use for authentication with the SMTP server. This property requires a value if **Authentication type** is **Basic authentication**. Minimum length: 0 Maximum length: 256

7. Click **OK** and **OK** again to save settings.

SMTP Adapter Property Schema and Properties

The following table lists the properties in the SMTP adapter property schema.

Namespace: http://schemas.microsoft.com/BizTalk/2003/smtp-properties

Name	Type	Description
Username	xs:string	Specify the user name to use for authentication with the SMTP server.
Password	xs:string	Specify the password to use for authentication with the SMTP server.
SMTPHost	xs:string	Specify the name of the SMTP server to use when sending messages.
From	xs:string	Specify the e-mail address to place on the SMTP **From** header.
CC	xs:string	Specify the e-mail address to send a copy of the message.
Subject	xs:string	Specify the subject header for the message.
SMTPAuthenticate	xs:int	Specify the type of authentication to use with the SMTP server.
ReadReceipt	xs:boolean	Specify whether to send a confirmation e-mail message when the message is read.
DeliveryReceipt	xs:boolean	Specify whether to send a confirmation e-mail

		message after delivery of the message.
EmailBodyText	xs:string	Specify text to be used for the body of the e-mail being sent.
EmailBodyTextCharset	xs:string	Specify the character set to use for encoding the body of the e-mail being sent when the **EmailBodyText** option is used. The SMTP adapter will convert the **EmailBodyText** to the character set specified by **EmailBodyTextCharset**.
EmailBodyFile	xs:string	Specifies that the contents of a file will be used for the body of the e-mail being sent and the full path to the file. This path must be accessible to the host for the SMTP adapter at run time.
EmailBodyFileCharset	xs:string	Specify the character set to use for encoding the body of the e-mail being sent if the **EmailBodyFile** property is set. The SMTP adapter will not perform any conversion on the file; the file must already be encoded in this character set. If the file has a Byte-Order-Mark (BOM), the SMTP adapter will remove it.
Attachments	xs:string	Specifies that a file or files will be attached to the e-mail message and the full path to the file or files. The specified path or paths must be accessible to the host for the SMTP adapter at run time.
MessagePartsAttachments	xs:int	Specify how BizTalk message parts are attached to the e-mail message
ReplyBy	xs:dateTime	Specify a dateTime value for the **Reply-To** header in the outgoing e-mail message.

Restrictions When Configuring the SMTP Adapter

This section contains information to consider when configuring the SMTP adapter.

In This Section

- Restrictions on Using Macros in SMTP Headers
- Restrictions on the SMTP To Property
- Restrictions on the SMTP Host Property

Restrictions on Using Macros in SMTP Headers

You can form the **Subject**, **To**, **From**, and **CC** properties on an SMTP message header dynamically by using a predefined set of macros. Before sending a message, the SMTP send handler substitutes all the macros in headers with their values. You can use several different macros when forming one header.

The SMTP send handler does not substitute macros in the **To**, **From**, or **CC** header if any of the following are true:

- The corresponding system property is not set.
- The macro is misspelled.
- The value for the macro contains symbols that are not valid for the SMTP headers.

If any of these conditions are met, the SMTP send handler leaves the macros as they are, for example, **%SourceParty%@somedomain.com** or **Message from %SourceParty%**.

The following table lists the macros you can use to build the **To**, **CC**, and **Subject** headers.

Macro	Description	For use with To	For use with CC	For use with Subject
%MessageID%	Globally unique identifier (GUID) of the message in BizTalk Server. The value comes from the message context property **BTS.MessageID**.	No	No	Yes
%datetime_bts2000%	UTC date time in the format YYYYMMDDhhmmsss, where sss means seconds and milliseconds (for example, 199707121035234 means 1997/07/12, 10:35:23 and 400 milliseconds).	No	No	Yes
%datetime%	UTC date time in the format YYYY-MM-DDThhmmss (for example, 1997-07-12T103508).	No	No	Yes
%datetime.tz%	Local date time plus time zone from GMT in the format YYYY-MM-DDThhmmssTZD, (for example, 1997-07-12T103508+800).	No	No	Yes
%time%	UTC time in the format hhmmss.	No	No	Yes
%time.tz%	Local time plus time zone from GMT in the format hhmmssTZD (for example,	No	No	Yes

	124525+530).			
%SourceParty%	Name of the source party from which the File adapter received the message.	Yes	Yes	Yes
%SourcePartyQualifier%	Qualifier of the source party from which the File adapter received the message.	Yes	Yes	Yes
%DestinationParty%	Name of the destination party. The value comes from the message context property **BTS.DestinationParty**.	Yes	Yes	Yes
%DestinationPartyQualifier%	Qualifier of the destination party. The value comes from the message context property **BTS.DestinationPartyQualifier**.	Yes	Yes	Yes

Restrictions on the SMTP To Property

The **To** property is a string that specifies the SMTP address of the recipient of the message. You can list several addresses with a separator that SMTP server supports.

There are no specific restrictions for the format of an SMTP address. This means that the adapter will accept any format that an SMTP server supports. If a user provides addresses that are not valid, the send operation will fail and the SMTP send adapter will report an error.

The only restrictions for this property are that it must not be empty and its size must not exceed 256 characters.

Restrictions on the SMTP Host Property

The SMTP host property is a string that specifies the SMTP server that the SMTP adapter will use to send messages from the BizTalk server.

The following rules and restrictions apply to this property:

- This property must be configured on the adapter handler level, on the endpoint level, or in both places.
- The SMTP server property cannot contain the following characters: ` ~ ! @ # $ ^ & * ( ) = + [ ] { } \ | ; : ' " , < > /, ?;
- The length of the SMTP server name must not exceed 256 characters.

The SMTP adapter always validates the SMTP host name at design time by using the previously mentioned rules. In addition, the SMTP adapter validates the SMTP host name at run time if a message is sent through a dynamic port with the SMTP adapter.

SMTP Adapter Security Recommendations

You use the SMTP adapter to exchange information between a server running BizTalk Server and other applications by means of the Simple Mail Transfer Protocol (SMTP) protocol. BizTalk Server can send messages to other applications by creating an e-mail message and delivering it to a specified e-mail address. You can use the SMTP adapter only for sending messages.

When you configure the service account for the host instance running the SMTP adapter, you need to specify the type of authentication you want to use with the remote SMTP server. The authentication options are basic authentication (clear text), NTLM (by using current credentials), or none if authentication is not required by the SMTP server.

When you send a message by using the SMTP adapter, BizTalk Server sends the message in clear text by default. If you use a pipeline that has an S/MIME encoder component, you can encrypt the message before you send it to the SMTP server. However, the SMTP header is still in clear text.

If you want to audit the e-mail messages that BizTalk Server sends, you should use the SMTP adapter to connect to your own SMTP server, where you can then audit the messages.

The SMTP adapter does not support Secure Sockets Layer (SSL).

SOAP Adapter

Microsoft BizTalk Server Web services use the SOAP adapter when receiving and sending Web service requests. The SOAP adapter enables you to publish orchestrations as Web services and consume external Web services.

In This Section

- What Is the SOAP Adapter?
- Configuring the SOAP Adapter
- SOAP Adapter Security Recommendations

What Is the SOAP Adapter?

Web services are programs with interfaces that adhere to the standards set forth in the Web Services Description Language (WSDL). You can use a Microsoft BizTalk Server orchestration to create and use Web services to combine separate but related business functions in a manageable and intuitive way. For more information about Web services and orchestration, The BizTalk Web Services Publishing Wizard is included with BizTalk Server 2006 to speed up the development process for publishing Web services. After you create an orchestration that you want to expose as a Web service, the BizTalk Web Services Publishing Wizard generates a Web service project for you. For more information about the BizTalk Web Services Publishing

Wizard, Note that prior to running the BizTalk Web Services Publishing Wizard, you must enable BizTalk Web services. For more information about enabling BizTalk Web services,.

The SOAP adapter consists of two adapters—a send adapter and receive adapter.

In This Section

- SOAP Receive Adapter
- SOAP Send Adapter
- Single Sign-On Support for the SOAP Adapter

SOAP Receive Adapter

You use the SOAP receive adapter to receive Web service requests. The SOAP receive adapter creates a BizTalk Message object, and promotes the associated properties to the message context.

SOAP Send Adapter

You use the SOAP send adapter to call a Web service. The SOAP send adapter reads the message context on the BizTalk Message object to get the proxy name and calls the associated external Web service proxy.

Client Authentication for the SOAP Send Adapter

The SOAP send adapter authenticates with the destination server by using one of the following authentication types:

- **Anonymous.** The default setting.
- **Basic.** The SOAP connection sends the user name and password in plain text.
- **Digest.** The SOAP connection sends the user name and password in an encrypted format.
- **Kerberos or NTLM.** Neither the user name nor the password is sent over a SOAP connection. The SOAP adapter always uses the credentials of the process under which the SOAP send adapter runs for this authentication type.

Additionally, the SOAP send adapter can provide a client Secure Sockets Layer (SSL) certificate to the Web server if the server requires or accepts it.

If you enabled Enterprise Single Sign-On (SSO), when the SOAP send adapter receives a message with the request to the **SSOTicket** property, the adapter connects to an SSO server to validate and redeem the ticket. After the SOAP adapter validates the ticket, it is decrypted and the credentials for the affiliate system are retrieved from the credential store. The SOAP

adapter then uses the credentials to connect to the affiliate system, and the SOAP request is processed.

Client Certificates for the SOAP Send Adapter

The SOAP send adapter can establish a secure connection with servers that accept or require client certificates. If you specify a client certificate, the SOAP send adapter uses the certificate when connecting with servers that require or accept client certificates. If you do not specify a client certificate and the destination server requires client certificates, the SOAP send adapter fails to send the message and follows the standard retry logic.

The SOAP send adapter uses the client certificate from the Personal store of the account under which the BizTalk Server process is running. The SOAP adapter specifies the certificate by its thumbprint. If the SOAP send adapter fails to load the certificate for any reason, the message that it was sending is suspended.

Negative Acknowledgement (NACK) Messages Generated for Failed Transmissions by the HTTP or SOAP Adapters

When a message is successfully transmitted, the BizTalk Messaging Engine publishes an associated Acknowledgement (ACK) message to the MessageBox database if delivery notifications are enabled. Likewise, when a message is suspended by the BizTalk Messaging Engine or an orchestration is suspended by the orchestration engine, BizTalk Server publishes an associated Negative Acknowledgement (NACK) message to the MessageBox. The NACK message contains context properties and a message body part consisting of a SOAP fault. If the NACK message is generated due to a failed transmission from the HTTP or SOAP adapters, the SOAP fault contains the **Headers** element and the **Body** element of the response from the destination Web server. The following is an example of the SOAP fault in a NACK generated for a failed SOAP transmission:

To subscribe to a NACK message, you can do one of the following:

1. Create a send port with a filter for the appropriate message context property. See **Message Context Properties** for a listing of system message context properties including those related to message acknowledgment.

2. Send from an orchestration port marked with **Delivery Notification = Transmitted**. If an orchestration port is marked with **Delivery Notification = Transmitted**, the orchestration will wait until it receives either an ACK or a NACK for the message that was transmitted. If a NACK is generated then it will be routed to the orchestration and the orchestration will throw a DeliveryFailureException. The DeliveryFailureException is deserialized from the SOAP fault that is contained within the NACK message body. To retrieve the exception message string from the SOAP fault that is returned to the orchestration, cast the DeliveryFailureException to a SoapException and then access the InnerXml from the SOAP Detail section.

Single Sign-On Support for the SOAP Adapter

You can use the BizTalk Server Administration console to configure Enterprise Single Sign-On (SSO) for use with the SOAP receive location or send port. This topic describes how SSO works with the SOAP adapter.

Single Sign-On Support for SOAP Receive Locations

SOAP receive locations support two versions of SSO—BizTalk Server 2006 Enterprise SSO and Microsoft SharePoint Portal Server SSO. Run the BizTalk Web Services Publishing Wizard to enable support for SharePoint Portal Server SSO. For more information about enabling SharePoint Portal Server SSO. Enable BizTalk Enterprise Single Sign-On by using the property pages for the SOAP receive location. For more information about enabling Enterprise SSO for the SOAP receive location, see **Configuring a SOAP Receive Location**.

Enterprise SSO Support for SOAP Receive Locations

Internet Information Services (IIS) receives a SOAP request from a Web client, and then IIS authenticates the user and passes the security identifier to the SOAP adapter. If the IIS authentication method is Digest authentication, Basic authentication, or Integrated Windows Authentication, the SOAP adapter calls the SSO credential store to obtain an encrypted ticket based on the authenticated user. This ticket is stored as the **SSOTicket** property in the context property of the message.

In the pass-through scenario, the BizTalk Messaging Engine directs the message to the MessageBox database. When a send adapter receives the message from the MessageBox database, it calls the RedeemTicket method with the encrypted ticket along with the application name to retrieve the security credentials for the application from the SSO store. The send adapter then uses the external credentials to connect to the application and process the request. For more information about the affiliate applications,

In scenarios where an orchestration invokes the send adapter, the BizTalk Messaging Engine sends the message to the MessageBox database. The orchestration should ensure that both the **SSOTicket** context property and the **Microsoft.BizTalk.XLANGs.BTXEngine.OriginatorSID** context property of the message that contains the ticket are maintained. When the adapter receives this message from the MessageBox database, the adapter calls the RedeemTicket method with the encrypted ticket to retrieve the back-end credentials from the SSO store. The user designing the orchestration should specifically copy this property to the message.

SharePoint Portal Server SSO Support for SOAP Receive Locations

When integrating with SharePoint Portal Server 2004, BizTalk Server supports the use of Microsoft SharePoint Portal Server SSO only through the SOAP adapter. SharePoint Portal Server creates SSO tickets and sends them to BizTalk Server in a SOAP header of the SOAP request. When the SOAP adapter receives a request containing an SSO ticket, the ticket is stored as the **SSOTicket** property in the context property of the message. This same property would contain an Enterprise SSO ticket. Only one SSO ticket can be associated with a BizTalk message.

In both pass-through and orchestration scenarios, the handling of an SSO ticket received from SharePoint Portal Server is the same as if the ticket were created by the SOAP adapter using Enterprise SSO. When a send adapter receives a message, it calls the **RedeemTicket** method with the encrypted ticket that SharePoint Portal Server generated. The send adapter does not need to be aware that different SSO tickets exist. The **RedeemTicket** method will determine which SSO system generated the ticket and redeem it from the appropriate place.

Combined Use of Enterprise SSO and SharePoint Portal Server SSO

BizTalk Server supports the simultaneous use of both SSO systems. The API can differentiate between the tickets generated by each SSO and will redeem them from the appropriate SSO database. If you use both SSO systems at the same time, the following rules determine which SSO ticket the SOAP receive location promotes to the **SSOTicket** context property:

- If neither SSO is enabled, do not promote a ticket.
- If the Enterprise SSO is enabled, but the SharePoint Portal Server SSO is not enabled, retrieve and promote the Enterprise SSO ticket.
- If the SharePoint Portal Server SSO is enabled, but the Enterprise SSO is not enabled, promote the existing SharePoint Portal Server SSO ticket.
- If both the Enterprise and SharePoint Portal Server SSO are enabled:
  - If the SharePoint Portal Server SSO ticket is received, promote that ticket.
  - If the SharePoint Portal Server SSO ticket is not received, retrieve and promote the Enterprise SSO ticket.

Single Sign-On Support for the SOAP Send Adapter

If SSO is enabled, when a SOAP send port receives a message with the **Secure** property (**SSOTicket**), it calls the SSO server to validate and redeem the ticket for an affiliate application. The administration application, affiliate administrators, or SSO administrators for the affiliate application can call SSO to redeem a ticket. SSO then decrypts the ticket and obtains the back-end credentials. The pass-through and orchestration scenarios described in the "Enterprise SSO Support for SOAP Receive Locations" section of the topic Single Sign-On Support for the SOAP Adapter are the same for the SOAP send port.

By default, the SOAP send port does not enable SSO. For more information about enabling SSO for the SOAP send port, see **Configuring a SOAP Send Port by Using BizTalk Explorer**.

Configuring the SOAP Adapter

This section describes how to configure a SOAP adapter.

In This Section

- How to Configure a SOAP Receive Handler
- **Configuring a SOAP Receive Location**
- How to Configure a SOAP Send Handler
- How to Configure a SOAP Send Port
- How to Configure a SOAP Send Port with a Remote BizTalk Management Database
- SOAP Adapter Configuration and Tuning Parameters
- SOAP Adapter Property Schema and Properties

How to Configure a SOAP Receive Handler

You can configure the SOAP receive handler settings by using the BizTalk Server Administration Console. If you configure the adapter using the BizTalk Server Administration Console, the handler override properties do not need to be set in BizTalk Explorer.

To change global variables for the SOAP receive handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.
2. In the expanded adapter list, click **SOAP**, in the right pane, right-click the receive handler that you want to configure, and then click **Properties**.
3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated, and then click **OK**.

How to Configure a SOAP Receive Location

You can configure a SOAP receive location either programmatically or by using the BizTalk Server Administration console.

How to Configure a SOAP Receive Location Programmatically

The BizTalk Explorer object model enables you to create and configure receive locations programmatically. The BizTalk Explorer object model exposes the **IReceiveLocation** receive location configuration interface that has a **TransportTypeData** read/write property. This

property accepts a SOAP receive location configuration property bag in the form of a name-value pair of XML strings. To set this property in the BizTalk Explorer object model, you must set the **InboundTransportLocation** property of the **IReceiveLocation** interface.

The **TransportTypeData** property of the **IReceiveLocation** interface does not have to be set. If it is not set, the SOAP adapter uses the default values for the SOAP receive location configuration as indicated in the following table.

The following table lists the configuration properties that you can set in the BizTalk Explorer object model for the SOAP receive location.

Property name	Type	Description
URI	String	Virtual directory containing the Web service on the deployment server.
AddressableURI	String	Public address field containing the entire, callable URL. Default value: Blank
UseSSO	Boolean	Specifies whether the SOAP adapter issues the Single Sign-On ticket to the messages that arrive on this receive location. Default value: False

Use the following format to set the properties:

The **URI** and **AddressableURI** properties are set using the **Address** and **PublicAddress** properties of the receive location object.

The following code fragment illustrates creating a SOAP receive location:

How to Configure a SOAP Receive Location with the BizTalk Server Administration Console

You can set SOAP receive location adapter variables in the BizTalk Server Administration console. If properties are not set in the receive location, the default receive handler values set in the BizTalk Server Administration console are used.

To configure variables for a SOAP receive location

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application you want to create a receive location in.

2. In the BizTalk Server Administration console, in the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

3. In the **Receive Port Properties** dialog box, in the left pane, select **Receive Locations**, and then in the right pane, double-click an existing receive location or click **New** to create a new receive location.

4. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **SOAP** from the drop-down list, and then click **Configure**.

5. In the **SOAP Transport Properties** dialog box, do the following:

Use this	To do this
Virtual directory plus Web Service .asmx file	Indicate the .asmx file created by the BizTalk Web Services Publishing Wizard. The format of this message is similar to the following: /PurchaseOrder/POOrchestration.asmx Where the full location of the .asmx file is http://localhost/PurchaseOrder/POOrchestration.asmx.
Public address	Specify the fully qualified URI for this receive location. The value for this property is a combination of the server name and the virtual directory. The specified URI should designate the public Web site URL for trading partners to connect to when sending messages to BizTalk Server. This information is optional and is not used by BizTalk Server. This parameter is available to allow administrators to document the public URL that the receive location is tied to.
Use Single Sign-On	Indicate that the SOAP adapter uses Enterprise Single Sign-On.

6. Click **OK**.

7. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and then click **OK** to save settings. For information about the **Receive Locations Properties** dialog box, see How to Create a Receive Location.

The security settings used by the SOAP receive location are set in IIS. By default, the SOAP receive location is not set to use anonymous authentication.

While the SOAP client calls the Web service, the SOAP adapter authenticates the SOAP client by using either Anonymous, Basic, Digest, or Windows Integrated authentication. If the user is verified, the user context is passed to the receive handler.

To update a virtual directory to use ASP.NET 2.0

1. Launch the Internet Information Services (IIS) Manager. Click **Start**, click **Programs**, click **Administrative Tools**, and click **Internet Information Services (IIS) Manager**.

2. If you need to connect to a remote IIS server, right click the **Internet Information Services** node and then click **Connect**.

3. Type the computer name for the remote IIS server and credentials if necessary.

4. Expand the server name that houses the Web site or virtual directory to be updated.

5. Expand **Web Sites**.

6. Expand the Web site to view the virtual directories under the Web site.

7. Right-click the virtual directory that you want to update to use ASP.NET 2.0 and then click **Properties**.

8. In the virtual directory properties dialog box, click the **ASP.NET** tab.

9. Click the drop-down option next to **ASP.NET version** and change it to 2.0 or later, and then click **OK** to apply changes.

How to Configure a SOAP Send Handler

Use the following procedure to configure the SOAP send handler.

To change global variables for a SOAP send handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **SOAP**, in the right pane, right-click the send handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.

4. On the **Proxy** tab, do the following.

Use this	To do this
Use proxy	Indicate whether the SOAP send handler uses a proxy server.
Server	Specify the name of the proxy server. This property only requires a value if **Use proxy** is selected.

	Type: String Minimum length: 0 Maximum length: 256
Port	Specify the port the SOAP send handler uses. This property only requires a value if **Use proxy** is selected. Default Value: 80 Type: Long Minimum value: 0 Maximum value: 65535
User name	Specify the user name to use for authentication. This property only requires a value if **Use proxy** is selected. Type: String Minimum length: 0 Maximum length: 256
Password	Specify the password to use for authentication. This property only requires a value if **Use proxy** is selected. Type: String Minimum length: 0 Maximum length: 256

5. Click **OK**.

How to Configure a SOAP Send Port

You can configure a SOAP send port either programmatically or by using the BizTalk Server Administration console.

How to Configure a SOAP Send Port Programmatically

The BizTalk Explorer object model exposes an adapter-specific interface for send ports named **ITransportInfo** that has the **TransportTypeData** read/write property. This property accepts a SOAP send port configuration property bag in the form of a name-value pair of XML strings. Note that to set this property in the BizTalk Explorer object model you must set the **OutboundTransportLocation** property of the **ITransportInfo** interface first.

The **TransportTypeData** property of the **ITransportInfo** interface is not required. If it is not set, the adapter uses the default values for the SOAP send port configuration, as indicated in the following table.

The following table lists the configuration properties you can set in the BizTalk Explorer object model for SOAP send ports.

Property name	Type	Description
URI	String	Virtual directory containing the Web service on the deployment server.
Username	String	User name to specify for accessing the target Web service. Default value: Blank
Password	String	User password to use for authentication with the server. Default value: Blank
ClientCertificate	String	Thumbprint of client SSL certificate. Default value: Blank
AffiliateApplicationName	String	The name of the SSO application to use to redeem the ticket for client credentials. The **AffiliateApplicationName** is mutually exclusive to a **Username** and **Password** pair. Default value: Blank
UseProxy	Boolean	Indicates whether the SOAP send port uses a proxy server to access the target Web service. Default value: False
ProxyAddress	String	Address of the HTTP proxy to use for the Web service call. Default value: Blank

ProxyPort	Integer	Port of the HTTP proxy to use for the Web service call. Default value: Blank
ProxyUsername	String	User name to use for the proxy. Default value: Blank
ProxyPassword	String	Password to use for the proxy. Default value: Blank

The following code shows the format to use to set these properties:

How to Configure a SOAP Send Port with the BizTalk Server Administration Console

You can set SOAP send port adapter variables in the BizTalk Server Administration console. If properties are not set for the send port, the default send handler values set in the BizTalk Server Administration console are used.

To configure variables for a SOAP send port

1. In the BizTalk Server Administration console, create a new send port or double-click an existing send port to modify it. See How to Create a Send Port for more information. Configure all of the send port options and specify **SOAP** for the **Type** option in the **Transport** section of the **General** tab.

2. On the **General** tab, in the **Transport** section next to **Type**, click **Configure**.

3. In the **SOAP Transport Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Web Service URL	Specify the address of the Web service you want to call.
Authentication	Indicate the authentication method used by the Web service you are calling. Options: • **Anonymous.** The default setting. • **Basic.** The SOAP connection sends the user name and password in plain text. • **Digest.** The SOAP connection sends the password in an encrypted format.

	• **NTLM.** Neither the user name nor the password is sent over a SOAP connection. The SOAP adapter always uses the credentials of the process under which the SOAP send adapter runs for this authentication type.
Credentials	Specify the type of credentials to use. Only available if the **Authentication type** is **Basic** or **Digest**. Options: • **Do Not Use Single Sign-On** **User name** The user name to use for authentication with the destination server. If the **Authentication type** property is **Anonymous** or **NTLM**, this option is disabled. This property requires a value if **Basic** or **Digest** is selected, and Enterprise Single Sign-On is not used. Minimum length: 0 Maximum length: 256 **Password** The password to use for authentication with the destination server. If the **Authentication type** property is **Anonymous** or **NTLM**, this option is disabled. This property requires a value if **Basic** or **Digest** is selected, and Single Sign-On is not used. Minimum length: 0 Maximum length: 256 • **Use Single Sign-On** Specify whether to use Single Sign-On to retrieve client credentials for authentication with the destination server. **Affiliate Application** Specifies the affiliate application to use for Single Sign-On. For information about populating this list, Minimum length: 0 Maximum length: 256
Client Certificate	Specify the thumbprint of the client certificate to use for establishing a

Thumbprint	connection. Example: 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF 01 23 45 67 Minimum length: 0 Maximum length: 59

4. In the **SOAP Transport Properties** dialog box, on the **Proxy** tab, do the following:

Use this	To do this
Use Handler's default proxy configuration	Specify the send port proxy handler configuration. This is the default setting.
Do not use proxy	Indicate whether the SOAP send handler uses a proxy server.
Use proxy	Indicate whether the SOAP send handler uses a proxy server.
Server	Specifies the name of the proxy server. This property only requires a value if **Use proxy** is selected. Type: String Minimum length: 0 Maximum length: 256
Port	Specify the port the SOAP send handler uses. This property only requires a value if **Use proxy** is selected. Default Value: 80 Type: Long Minimum value: 0 Maximum value: 65535
User name	Specify the user name to use for authentication. If you use Windows integrated authentication, the user name includes the domain, **domain\username**. If you use Basic or Digest authentication, the user name does not include **domain**.

	This property only requires a value if **Use proxy** is selected. Type: String Minimum length: 0 Maximum length: 256
Password	Specify the password to use for authentication. This property only requires a value if **Use proxy** is selected. Type: String Minimum length: 0 Maximum length: 256

5. In the **SOAP Transport Properties** dialog box, on the **Web Service** tab, do the following:

Use this	To do this
Orchestration Web Port	Specify to use the Web service that is exposed at the Web Service URL listed on the **General** tab. This is the default setting.
Assembly name	Specify the name of the assembly containing the Web service proxy. This field can be populated by clicking the browse button to find an assembly. After selecting the assembly this box is populated with the fully qualified name of the assembly.
Type name	Specify the name of the class that contains the Web method to be invoked. This can be selected from a list of types contained within the assembly.
Method name	Specify one of the methods in the list box or choose the option to "Specify later". If the option to "Specify later" is chosen, the Web method must be set by some other means, such as a pipeline component. In this scenario, the web method must be written to the Soap Adapter **MethodName** context property.
SOAP 1.2	Specify to generate proxy code that will support the SOAP 1.2 protocol. If this option is left cleared, SOAP 1.1-compliant proxy code will be generated. Default value: cleared

6. Click **OK** and **OK** again to save settings.

How to Configure a SOAP Send Port with a Remote BizTalk Management Database

The SOAP send port has port-level configuration that may require a user name and password. When you export this configuration to a binding file, the system does not keep the password in the binding file for security reasons. As a result, in cases when you configure user names and passwords in the source environment, after you import the binding file into the destination environment (with a different BizTalk Management database), you must use BizTalk Explorer (or the BizTalk Explorer object model) to specify passwords for these port configurations. If the destination environment does not have Microsoft Visual Studio 2005 installed, use the following steps to complete the SOAP send port configuration for the destination environment in BizTalk Explorer.

To configure a SOAP send port with a remote BizTalk Management database

1. Click **Start**, point to **Programs**, point to **Microsoft Visual Studio 2005**, and then click **Microsoft Visual Studio 2005**.

2. On the **View** menu, click **BizTalk Explorer**.

3. In BizTalk Explorer, right-click the **BizTalk Configuration Databases** node, and then click **Add Database**.

4. In the **Add Database** dialog box, specify the management database of the production environment. A new node appears under the root node.

5. Expand the newly added node, select the send port that needs reconfiguration, and specify a password for the send port.

SOAP Adapter Configuration and Tuning Parameters

You can configure the number of concurrent connections that the SOAP adapter opens for a particular destination server by making an entry in the BTSNTSvc.exe.config file that is located in the root BizTalk Server installation directory.

SOAP Adapter Property Schema and Properties

The following table lists the properties in the SOAP adapter property schema.

Namespace: http://schemas.microsoft.com/BizTalk/2003/soap-properties

Name	Type	Description
AssemblyName	xs:string	Identifies the .NET type and assembly to be loaded and executed.
MethodName	xs:string	Identifies the target method on the .NET assembly

		that is to be invoked.
Username	xs:string	User name to use for authentication with the server.
Password	xs:string	User password to use for authentication with the server.
ClientCertificate	xs:string	Thumbprint of the client SSL certificate.
UseProxy	xs:Boolean	Specifies whether the SOAP adapter uses a proxy server.
ProxyAddress	xs:string	Specifies the proxy server address.
ProxyPort	xs:int	Specifies the proxy server port.
ProxyUsername	xs:string	Specifies the user name for authentication with the proxy server.
ProxyPassword	xs:string	Specifies the user password for authentication with the proxy server.
UnknownHeaders	xs:string	Specifies the serialized list of unknown SOAP headers.
AffiliateApplicationName	xs:string	Defines the name of the affiliate application to use for SSO.
AuthenticationScheme	xs:string	Specifies the type of authentication to use with the destination server.
UseSSO	xs:boolean	Specifies whether the SOAP adapter uses SSO for the send port.
UseHandlerSetting	xs:boolean	Specifies whether the SOAP send port uses the proxy configuration for the handler.
ClientConnectionTimeout	xs:int	Specifies the time-out period of waiting for a response from the server. If set to zero (0), the system will calculate the time-out based on the request message size.
UserDefined	xs:string	Defines user-defined classes.
UseSoap12	xs:boolean	Specifies whether to generate proxy code that supports the SOAP 1.2 protocol.

SOAP Adapter Security Recommendations

BizTalk Server uses the SOAP adapter to publish (receive) and consume (send) Web services. For more information about the SOAP adapter, It is recommended you follow these guidelines for securing and deploying the SOAP adapter in your environment.

- For security recommendations for publishing Web services.

- The SOAP adapter leverages the Hypertext Transfer Protocol (HTTP) to send and receive messages to and from BizTalk Server. Therefore, you must follow the security recommendations for securing Internet Information Services (IIS). If you use IIS 6.0, ensure you follow the IIS 6.0 recommendations for configuring application isolation. For more information, see the Microsoft TechNet Web site at http://go.microsoft.com/fwlink/?LinkId=25222. If you use IIS 5.0 or 5.1, ensure you follow the IIS 5.0 recommendations for securing IIS 5.0. For more information, see the Microsoft TechNet Web site at http://go.microsoft.com/fwlink/?LinkId=24776.

- When you create an application pool for a SOAP receive location, you must configure it to run under an account that is a member of the Windows group for the isolated host running the SOAP receive adapter and the Internet Information Services Worker Process group (IIS_WPG group). You must then configure the host instance for the SOAP receive adapter to use this account. If you change the account for the IIS_WPG group, you must ensure you also update the host instance to run under the new account.

- When you use Secure Sockets Layer (SSL) client certificates with the SOAP send adapter, you must manually configure these certificates. For more information about configuring the SSL client certificates, see **Configuring a SOAP Send Port by Using BizTalk Explorer**.

- When consuming Web services, you can use anonymous, basic, digest, Windows integrated, or client certificates for authentication. When consuming Web services by using basic authentication, it is recommended to use SSL to ensure that an unauthorized person cannot read the user credentials from the message.

- You can use Enterprise Single Sign-On (SSO) in scenarios where you need to map the content of the front-end user to credentials in a back-end system. For more information,

- When using basic authentication, or when you do not use encryption at the message level, it is recommended to use SSL for both receiving and sending messages to ensure that an unauthorized person cannot read the user credentials.

- It is recommended to use Windows integrated authentication for both sending and receiving messages.

- The computer running the SOAP adapter also has the BizTalk Server runtime. It is recommended you do not put the SOAP adapter in the perimeter network. If you do, you have to open ports from the perimeter network to the data domain for SQL Server traffic to the MessageBox database, and you are exposing the BizTalk Server runtime to potential attacks. It is recommended you configure the SOAP adapter in the processing domain

(that is, not the perimeter network). You can then configure the outermost firewall to forward SOAP requests through the firewall in the processing domain. This mechanism is called reverse proxy. (The ISA implementation is called Web Publishing.)

SQL Adapter

The SQL adapter exchanges data between Microsoft BizTalk Server and a SQL Server database. You can use the SQL adapter to poll data from one or more data tables and transmit the data as one or more XML messages to BizTalk Server. You can also use the SQL adapter to move large amounts of data to or from the SQL Server database as part of a BizTalk Server messaging or orchestration solution. In addition, you can use the SQL adapter to insert, update, and delete data in SQL Server tables by using SQL updategrams or by invoking stored procedures.

In This Section

- What Is the SQL Adapter?
- Configuring the SQL Adapter
- Using the SQL Adapter

What Is the SQL Adapter?

The SQL adapter consists of two adapters—a receive adapter and a send adapter.

This section discusses the workflow for both the SQL receive adapter and the SQL send adapter.

In This Section

- SQL Receive Adapter
- SQL Send Adapter

SQL Receive Adapter

The SQL receive adapter is a polling adapter that periodically polls for SQL result sets. An SQL command is stored in the receive location configuration properties and is invoked by the Adapter Framework scheduler. The SQL receive adapter supports **SELECT** statements and stored procedure calls that each return single result sets in the form of XML data.

SELECT Statements

The SQL receive adapter supports SQL **SELECT** statements that contain static values in the **WHERE** clause. Examples of supported **SELECT** statements are:

The following examples illustrate the supported SQL clauses that determine how the SQL receive adapter returns XML data:

- `xml auto`. Supported
- `xml auto, elements`. Supported.
- `xml raw`. Not supported

The first clause, `xml auto`, is required to return data in XML format, with each element containing the table name as the element name and the column data returned as attribute values. For example, the statement `Select * from customers for xml auto` returns data in the following format:

The second clause, `xml auto, elements` is similar to `xml auto` but returns column data as child elements instead of attribute values. For example, the statement `Select * from customers for xml auto, elements` returns data in the following format:

The third clause, `xml raw`, is not supported by the SQL receive adapter.

Stored Procedures

The SQL receive adapter supports SQL stored procedure calls that contain static parameter values. An example of a supported stored procedure call is as follows:

When using stored procedure calls in the SQL receive adapter, the stored procedure should update the retrieved records so that the adapter does not retrieve them during the next polling cycle. For example, consider the following stored procedure:

This stored procedure would return all rows that contain a null value for the ShippedDate every time the SQL receive adapter calls it. Modify this stored procedure to update rows after the adapter retrieves them:

When using the Add Generated Items Wizard to add SQL schemas to a BizTalk project, modify the **SELECT** statement in the stored procedure to contain the **xmldata** clause. This clause informs SQLXML to return the schema that represents the result set to the wizard. Change the stored procedure to the following:

After completing the wizard, it is important to remove the **xmldata** clause from the **SELECT** statement. Not doing this will cause the SQL receive adapter to return the metadata along with any result set data. Note that when you do not specify the **xmldata** clause and the **SELECT** statement or stored procedure does not return any results, the SQL adapter does not submit a document to the BizTalk messaging subsystem. However, the SQL adapter always submits a document if the **xmldata** clause is present in the **SELECT** statement or stored procedure.

SQL Send Adapter

You use the SQL send adapter to send dynamically created updategrams or dynamically invoked stored procedures to SQL Server. An updategram is an XML fragment that inserts, updates, or deletes data in a SQL Server database by mapping XML nodes against database

tables and columns. SQL Server returns an optional response document after the updategram completes, which contains the success status of the update. If a failure occurs during the update, the SQL adapter throws an exception that the BizTalk Messaging Engine handles. When the SQL send adapter is configured to invoke a stored procedure, it returns any results in the form of a single XML-formatted record set.

Using Updategrams

All updategrams contain the same basic structure:

The following definitions describe the role of each block:

- **<before>.** Identifies the existing state (also referred to as "the before state") of the record instance, and works as the **WHERE** clause in an SQL statement.

- **<after>.** Identifies the new state to which data is to be changed.

- **<sync>.** Contains the <before> and <after> blocks. A <sync> block can contain more than one set of <before> and <after> blocks. If more than one set of <before> and <after> blocks exist, you must specify these blocks (even if they are empty) as pairs. Furthermore, an updategram can have more than one <sync> block. Each <sync> block is one unit of transaction (which means that either everything in the <sync> runs or nothing runs). If you specify multiple <sync> blocks in an updategram, the failure of one <sync> block does not affect the other <sync> blocks.

Whether an updategram deletes, inserts, or updates a record instance depends on the contents of the <before> and <after> blocks:

- If a record instance appears only in the <before> block with no corresponding instance in the <after> block, the updategram performs a delete operation.

- If a record instance appears only in the <after> block with no corresponding instance in the <before> block, it is an insert operation.

- If a record instance appears in the <before> block and has a corresponding instance in the <after> block, it is an update operation. In this case, the values specified in the <after> block update the record instance.

The Add Generated Items Wizard enables you to select which operation to perform on the table and generates a schema that supports only that operation. For example, if you select an insert operation, only the <before> block on the updategram will appear in the schema. If you want to generate a multipurpose schema that you can use for any of these operations, select an update operation with all the columns from the **Columns to Update** list. Then you can select which blocks to create when creating XML instances from this schema.

The schemas created by the Add Generated Items Wizard are not exact replicas of updategrams. These schemas contain the unique root element names provided while running the wizard as well as a few other modifications designed to improve usability between BizTalk Server and SQLXML. For more information about updategrams, see the section titled "Using Updategrams to Modify Data" in the SQLXML 3.0 Service Pack 1 documentation.

Insert Operations

Updategrams that perform insert operations contain only the <after> block:

The attribute data in the table name element represents the column names and values. The Add Generated Items Wizard for the SQL adapter enables you to select which columns to include in the insert operation. The wizard adds each column chosen in the **Columns to Choose** box to the schema, which can be set to a value when creating instances of the insert updategram.

Update Operations

Updategrams that perform update operations contain both the <before> and <after> blocks:

The data columns in the <before> block are used as the **WHERE** clause in the update operation. The Add Generated Items Wizard for the SQL adapter enables you to select which columns to include in the <before> block, and automatically adds all of the table columns to the <after> block. Each column chosen in the **Columns to Choose** list is added to the <before> block.

Delete Operations

Updategrams that perform delete operations contain only the <before> block:

The data columns in the <after> block are used as the **WHERE** clause in the delete operation. The Add Generated Items Wizard for the SQL adapter enables you to select which columns to include in the <after> block in the **Columns to Choose** list.

Using Stored Procedures

See the BizTalk Server 2006 SDK sample Using the SQL Adapter with a Stored Procedure in an Orchestration for a detailed walkthrough of how to use a stored procedure with the SQL send adapter.

Configuring the SQL Adapter

This section describes how to configure a SQL adapter.

In This Section

- How to Configure a SQL Receive Handler
- How to Configure a SQL Receive Location
- How to Configure a SQL Send Handler
- How to Configure a SQL Send Port
- How to Add SQL Adapter Schemas to a BizTalk Project

How to Configure a SQL Receive Handler

Use the following procedure to change the host associated with the SQL receive handler.

To configure the general properties for a SQL receive handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **SQL**, in the right pane, right-click the receive handler that you want to configure, and then click **Properties**.

3. In the **<host name> Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the receive handler will be associated.

4. Click **Properties**.

5. In the **SQL Transport Properties** dialog box, on the **Properties** tab, in the **Adapter Properties** section, edit the value for the **Error Threshold** property if necessary, and then click **OK**. The **Error Threshold** is used to specify the maximum number of continuous errors received before disabling the receive handler.

6. Click **OK**.

How to Configure a SQL Receive Location

You can set SQL receive location adapter variables either programmatically or by using the BizTalk Server Administration console.

How to Configure a SQL Receive Location Programmatically

The SQL adapter stores its configuration information in the Credential database. You can set this configuration information programmatically by using the BizTalk Explorer object model. The BizTalk Explorer object model exposes the **IReceiveLocation** configuration interface that contains the **TransportTypeData** read/write property. This property accepts the SQL receive location configuration property bag in the form of a name/value pair XML string.

The **TransportTypeData** property of the **IReceiveLocation** interface does not have to be set. If it is not set, default values for the SQL receive location configuration are used.

The following table lists the configuration properties that you can set for a SQL receive location.

Property name	Type	Description	Restrictions	Comments
sqlCommand	String	Specify the **SELECT** statement or stored procedure used when polling SQL Server for data.	String Required	None
connectionString	String	Specify the connection string to use to connect to an SQL database.	String Required	None
documentRootElementName	String	Specify the root element name used in the XML document received from SQL Server.	String Required	None
documentTargetNamespace	String	Specify the target namespace used in the XML documents received from SQL Server.	String Required	None
pollingUnitOfMeasure	String	Specify the unit of measure used between polling requests.	Integer Valid values are Seconds, Minutes, and Hours	If not set the default value is set to Seconds
pollingInterval	Long	Specify the number of units between polling requests.	Integer Minimum value: 1 Maximum value: Max Int32	If not set the default value is set to 30
pollWhileDataFound	Boolean	Specify whether to submit additional batches until the stored procedure or query returns no results, or submit a single stored procedure or query result for each polling interval.	Boolean	If not set the default value is set to **False**

The following code shows the format of the XML string you use to set the properties:

How to Configure a SQL Receive Location with the BizTalk Server Administration Console

You can set SQL receive location adapter variables in the BizTalk Server Administration Console.

To configure a SQL receive location with the BizTalk Server Administration console, use the following procedure.

To configure per instance variables for a SQL receive location

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application you want to create a receive location in.

2. In the BizTalk Server Administration Console, in the left pane, click the **Receive Port** node. Then in the right pane, right-click the receive port that is associated with an existing receive location or that you want to associate with a new receive location, and then click **Properties**.

3. In the **Receive Port Properties** dialog box, in the left pane, select **Receive Locations**, and then in the right pane, double-click an existing receive location or click **New** to create a new receive location.

4. In the **Receive Location Properties** dialog box, in the **Transport** section next to **Type**, select **SQL** from the drop-down list, and then click **Configure**.

5. In the **SQL Transport Properties** dialog box, do the following:

Use this	To do this
Poll While Data Found	Specify whether to submit additional batches until the stored procedure or query returns no results, or submit a single stored procedure or query result for each polling interval. Default value: False
Polling Interval	Specify the number of units between polling requests. Default value: 30
Polling Unit of Measure	Specify the unit of measure used between polling requests. Default value: Seconds Valid values: Seconds, Minutes, Hours
Connection	Specify the connection string to use to connect to an SQL database. For

String	example, Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;User ID=useracct;Initial Catalog=BTS2006_SQL_Adapter_loans;Data Source=localhost. You can also click the ellipses (**...**) button to build a connection string using the **Data Link Properties** dialog box.
Document Root Element Name	Specify the root element name used in the XML document received from SQL Server.
Document Target Namespace	Specify the target namespace used in the XML documents received from SQL Server.
SQL Command	Specify the **SELECT** statement or stored procedure used when polling SQL Server for data. For example: or Type this information into the box, or if you deployed the document schema for this receive location, you can import this information from the schema. To import the schema, click the ellipsis button to open the **Import information from a generated schema** dialog box, select the name of the project and schema, and then click **OK**. The **SQL Command** along with the **Document Target Namespace** and **Document Root Element Name** will be filled in.
URI	Identifies the receive location using the server and database name. The URI should take the form **SQL:/ / <** *DBServerName* **> / <** *DBName* **> [/ <** *string for identity purpose* **>]** where **SQL:/ /** is the prefix for invoking the SQL adapter, *DBServerName* is the server name for the target database (use "." for local), and *DBName* is the target database name, for example, Northwind. Any strings after the *DBName* are optional. Example: **SQL:/ /./Northwind** or **SQL:/ /MySQLServer001/MyTestDatabase/DynamicTestPort**

	Default value: SQL://

6. Click **OK**.

7. In the **Receive Location Properties** dialog box, enter the appropriate values to complete the configuration of the receive location, and then click **OK** to save settings. For information about the **Receive Location Properties** dialog box,

How to Configure a SQL Send Handler

You can configure the SQL send handler by using the BizTalk Server Administration Console. The dynamic SQL send ports use all of the properties set on the send handler.

To configure a SQL send handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, click **SQL**, in the right pane, right-click the send handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the send handler will be associated.

4. Click **Properties**.

5. In the **SQL Transport Properties** dialog box, on the **Properties** tab, do the following.

Use this	To do this
Connection String	Optional. Specify the connection string to use to connect to an SQL database. You can click the ellipsis (**...**) button to build a connection string using the **Data Link Properties** dialog box.
Document Target Namespace	Optional. Specify the target namespace used in the XML documents sent and received (in the case of Solicit-Response Port) from SQL Server.
Response Root Element Name	Optional. Specify the root element name contained in the XML documents received from SQL Server.

6. Click **OK**, and then click **OK** again to close the **Adapter Handler Properties** dialog box.

How to Configure a SQL Send Port

You can set SQL send port adapter variables in the BizTalk Server Administration Console.

To configure per instance variables for a SQL send port

1. In the BizTalk Server Administration Console, create a new send port or double-click an existing send port to modify it. See How to Create a Send Port for more information. Configure all of the send port options and specify **SQL** for the **Type** option in the **Transport** section of the **General** tab.

2. On the **General** tab, in the **Transport** section, next to **Type**, click **Configure**.

3. In the **SQL Transport Properties** dialog box, do the following:

Use this	To do this
Connection String	Specify the connection string to use to connect to an SQL database. You can click the ellipsis (**...**) button to build a connection string using the **Data Link Properties** dialog box.
Document Target Namespace	Specify the target namespace used in the XML documents sent and received (in the case of Solicit-Response Port) from SQL Server.
Response Root Element Name	Specify the root element name used in the XML documents received from SQL Server. This property may be empty for a one-way port.

4. Click **OK** and **OK** again to save settings.

How to Add SQL Adapter Schemas to a BizTalk Project

The Adapter Framework provides the means to add adapter schemas to BizTalk projects. For the SQL adapter, this requires the user to select either an existing endpoint (receive location or send port), or use the **Data Link Properties** dialog box to select a server to connect to. Then the user needs to enter the information used to generate the schema. When the wizard completes, two schemas are added to the BizTalk project if a send port is used and one schema is added to the project if a receive port is used.

The Add Adapter Wizard enables you to add a SQL adapter to a BizTalk project.

To add a SQL adapter to your project

1. In your Microsoft Visual Studio 2005 BizTalk project, in Solution Explorer, right-click your project, click **Add**, and then click **Add Generated Items**.

2. In the **Add Generated Items - <***Project name***>** dialog box, in the **Templates** section, select **Add Adapter**, and then click **Open**.

3. In the Add Adapter Wizard, on the **Select Adapter** page, do the following.

Use this	To do this
Adapter	Select **SQL**.
SQL Server	Type the BizTalk Server database name or select it from the drop-down list. This property is optional. If left blank, you must configure it later in BizTalk Explorer.
Database	Displays the list of BizTalk Management databases for the selected server. This property is optional. If left blank, you must configure it later in BizTalk Explorer.
Port	Displays a list of SQL send ports and receive locations previously created and stored in the BizTalk Management database. If selected, this port appears in the **Connection String** box on the next page in the wizard. This property is optional.

4. Click **Next**.

5. On the **Database Information** page, do the following.

Use this	To do this
Set button	Set the connection string information. For information about the **Data Link Properties** dialog box, see Step 6.
Connection String	If you chose a port on the previous page, the connection string information for that port automatically appears in this box. If your connection string is correct, click **Next** and proceed to step 8.

6. In the **Data Link Properties** dialog box, on the **Connection** tab, do the following.

Use this	To do this
Select or enter a server name	Type the name of the server or select a server name from the list. Use the **Refresh** button to update the list of servers.
Use Windows NT Integrated Security	Use Microsoft Windows NT integrated security when connecting to the SQL Server computer.
Use a specific user name and password	Connect to the SQL Server computer using non-integrated security.
Select the database on the server	Select the SQL Server database to which you want to connect.

Attach a database file as a database name	Type the name of the database.
Using the file name	Browse to the database to which you want to connect.
Test Connection	Test the connection to the SQL Server computer.

7. Click **OK**, and then return to step 5.

8. On the **Schema Information** page, do the following.

Use this	To do this
Target Namespace	Type the target namespace of the XML documents extracted from the SQL Server.
Receive Port	Select this option for a receive port.
Send Port	Select this option for a send port.
Document root element name	This option is only available when creating a receive port. Type the name of the input root element. The element name must be a valid XML element name. This element designates the document going to the SQL Server computer.
Request root element name	This option is only available when creating a send port. Type the name of the input root element. The element name must be a valid XML element name. This element designates the document going to the SQL Server computer.
Response root element name	This option is only available when creating a send port. Type the name of the output root element. The element name must be a valid XML element name. This element designates the document coming from the SQL Server computer.

9. Click **Next**.

   If you selected **Receive Port**, proceed to step 10. If you selected **Send Port**, continue to step 13.

10. On the **Statement Type Information** page, do the following.

Use this	To do this
Select Statement	Run the SQL statement.
Stored Procedure	Run the stored procedure.

11. Click **Next**.

    If you selected **Select Statement**, continue to step 12. If you selected **Stored Procedure**, continue to step 14.

12. On the **Statement Information** page, do the following.

Use this	To do this
SQL Script	Run an SQL statement. Because the SQL adapter uses SQLXML to render result sets as XML, the **for xml auto** clause should be included in the **SELECT** statement.

13. Click **Next**, and then continue to step 18.

14. On the **Statement Information** page, do the following.

Use this	To do this
Stored Procedure drop-down list	Select the stored procedure to run from the drop-down list.
Parameter values	Click a cell in the **Value** column twice (in two different locations) to enter a value for the property.
Generate	Click this button after you have entered values for all of the properties. This populates the **Generated Script** field.

15. Click **Next** and then continue to step 20.

16. On the **Statement Type Information** page, do the following.

Use this	To do this
Updategram	Run an updategram.
Stored Procedure	Run the stored procedure.

17. Click **Next**.

If you selected **Updategram**, continue to step 18. If you selected **Stored Procedure**, go to step 14.

18. On the **Statement Information** page, do the following.

Use this	To do this
Insert	Select this option to create an insert schema.
Update	Select this option to create an update schema.
Delete	Select this option to create a delete schema.
Table name	Lists the tables available through the connection string.
Columns to update	Filled with an enumerated list of table columns from the selected table. The behavior of this field depends on the selection of the **Insert**, **Update**, and **Delete** properties: • **Insert.** Every single column becomes part of the schema that inserts data into the table. Select all the primary keys, unless they are ID fields. Primary keys have **PK** in the Note column. • **Update.** All columns are available for update. The selected columns will become part of the **WHERE** clause. If more than one row meets the update criteria, the update operation will fail. • **Delete.** The selected columns will become part of the **WHERE** clause. If more than one row meets the delete criteria, the delete operation will fail.

19. Click **Next**.

20. Click **Finish** to complete the wizard.

An XSD schema that describes the messages that the SQL adapter sends and receives is now in the current project. Additionally, an orchestration schedule is now included in the project. The orchestration describes the port types and operations. You can use the orchestration as a template, or in the case where only messaging is used (content-based routing), you can remove the orchestration from the project. You can now build and deploy the project to deploy the schema.

Using the SQL Adapter

The content in this section describes how to configure multiple SQL adapter ports in an orchestration and enumerates recommended best practices for using the SQL adapter.

In This Section

- Using Multiple SQL Receive Adapter Ports
- Best Practices for Using the SQL Adapter
- Permissions and Database Object Names
- SQL Adapter Security Recommendations

Using Multiple SQL Receive Adapter Ports

The URI for SQL adapter receive ports is normally of the form **SQL://host/database**. However, if you create two receive ports on the same database, the second receive port will issue an error that the URI is already in use.

The workaround is to edit the URI in the receive location to be of the form **SQL://host//database/ID**, where ID is anything added to the URI. The example the SQL Adapter Sample uses is **SQL://localhost/BTS2004_SQL_Adapter_Loans/Report**.

To access the URI property

1. In the BizTalk Server Administration console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, expand **<All Artifacts>**, and click to select **Receive Locations**.
2. Double-click the receive location in the right pane to display the **Receive Location Properties** dialog box.
3. Click the **Configure** button on the **General** page of the **Receive Location Properties** dialog box to display the **SQL Transport Properties** dialog box.
4. In the **SQL Transport Properties** dialog box, append a unique identifier to the URI.

Best Practices for Using the SQL Adapter

The following are best practices to follow when using the SQL adapter:

- Have SQL Server do most of the work inside stored procedures:
  - Stored procedures are compiled and optimized compared to the interpreted updategrams. If there are multiple operations to database tables, the stored procedure will do all the work inside of SQL Server rather than having BizTalk Server form and manipulate multiple messages through updategrams.
  - The other side is that updategrams do not place any new code in the SQL Server computer. For an occasional operation (like updating the Interest field for a Loan in this sample), the updategram is adequate.

- Avoid competition in receive ports resubmitting the same data:
  - The SQL adapter receive port issues the same SQL operation (a **SELECT** or an **EXEC**) at regular intervals. A good practice is to have a state field in the table to prevent the SQL operation from reissuing the same information to BizTalk Server if the orchestration is taking a long time. In this example, the Loans table has a Status column. When created, the loan will have Status='new'. The stored procedure **SP_monitor_for_loan_to_assign()** will look only for Loans with Status = 'new' and immediately update the row to Status='in-use' to prevent resubmitting the information. To do this effectively, perform the state management of the row in a stored procedure as in this sample.
- Balanced use of "TOP 1" operations:
  - It is possible to have the returned documents return multiple rows or only one row, using **SELECT TOP 1**. Knowing that the returned documents have one subelement simplifies any mapping and message assignment shapes.
  - SQL adapter receive functions are limited to 60 messages per minute, unless the **Poll While Data Found** property is set to **True**, in which case the adapter will continuously run the SQL operation until no data is returned. If you need a higher throughput, you may need to return multiple rows at a time and use **Loop** constructs to manipulate the data. Note that you should ensure that the returned datasets are relatively small.

Improving Performance

Follow this recommendation to improve performance:

- If a large message, greater than 1 MB, is submitted from SQL Server to BizTalk Server, you should consider reducing the number of rows that are returned from SQL Server to keep the message under 1 MB. Setting the continuous polling property to **True** so that the adapter only submits the remaining rows in separate batches is also a recommended practice. The adapter submits the batches until the SQL Server computer returns an empty result set, at which point the SQL adapter releases the thread until the next polling interval.

Tips for Working with Currency

The following sections provide tips for working with currency.

Support for Currency Types

The following applies:

- Stored procedures do not support Money and Smallmoney types.
- Euro € currency and Czech currency symbols are not supported with updategrams.

Supporting the euro (€) Symbol When Using the SQL Adapter

Due to a limitation in SQLXML 3.0 SP3, the SQL adapter does not support some currency types when using updategrams. Updategrams will work for most currency types, but some currency types, such as the euro, will not work.

The following sample shows how you might expect the SQL adapter to work with the euro.

Sample table

Example of an updategram to insert a row in the table

Workaround

The workaround is to use a stored procedure and pass money values as strings, as shown in the following steps:

1. Create a stored procedure to wrap the insert:

2. Document the sample to call the stored procedure in step 1:

Mapping Between SQL Types and XML Types

Follow these guidelines for mapping between SQL types and XML types:

- The mapping between the SQL type money and the XML data types can have problems if the user uses the dollar or euro symbols (for example, <RequestedAmount>$100.45</RequestedAmount>). BizTalk Server will have to handle the money amount as a string. When passed into the SQL database, use 'CONVERT(money, @value)'. Note that SQL Server will lose monetary symbols such as $ or € if mixed money values are used. Consider adding a column to the tables denoting the type of money tendered.

- The SQL money type is converted to an XSD string type in BizTalk Server. When creating documents that have columns of SQL type money, you must prepend the dollar sign ($).

Tips for Using Updategrams

The following sections provide tips for working with binary type data and bit types, and tips for adding <before> and <after> blocks to updategrams.

Working with Binary Type Data

If a table has binary type data, it has to have a primary key to generate schemas for updategrams. If a table has no key, use a stored procedure instead of an updategram.

Working with Bit Types

Adding <before> and <after> Blocks to Updategrams

The default updategram schema only allows one <before> and one <after> block (minOccurs="0" maxOccurs="1"), but multiple pairs of <before> and <after> blocks are also legal for other updategrams.

For example, the following data instance is not legal for the updategram default schema, but it is a legal updategram:

For insert and delete, the default schema only allows one insert inside <before> or <after> blocks (minOccurs="1" maxOccurs="1"), but multiples are allowed in updategrams.

For example, the following data instance is not legal for schema, but it is a legal updategram:

Tips for Working with Stored Procedures

The following are tips for working with stored procedures:

- Save `Tran` and `RollBack tran` in stored procedures will cause an error event from the SQL adapter, but the transaction does roll back.
- SQLXML will associate elements and attributes of the body with an arbitrary namespace. When decomposing the document, it may be necessary to transform the namespace to something more desirable, such as the original root element.
- For stored procedures, the generated root element name is the one specified in the root element edit text control in the Add Adapter Wizard.

Permissions and Database Object Names

This topic contains useful considerations about the SQL adapter:

- A time-out value cannot be set when performing operations using the SQL adapter. Operations that fail will time out after approximately 30 seconds.

Setting Appropriate Permissions

Following these guidelines for setting permissions:

- Only give access to SQL database tables to those accounts that require access, and provide them with minimum privileges.
- Permissions required to modify SQL tables:
  - To perform an **INSERT** to a table, the user must have **INSERT** permissions.

- To perform an **UPDATE** to a table, the user must have both **UPDATE** and **SELECT** permissions.

- To perform a **DELETE** to a table, the user must have both **DELETE** and **SELECT** permissions.

- In some cases, the Add Adapter Wizard needs to access system tables, such as sysObjects and sysColumns, to get a list of tables, columns, or parameters. If the user does not have sufficient privileges to access those tables on the **Database Information** page of the Add Adapter Wizard, the wizard will fail. For example, to generate an updategram send schema on a target table, it is not sufficient if a user has only read rights on the target table.

Restrictions on Database Object Names

The following restrictions apply to database object names:

- SQLXML encodes certain characters automatically. SQL Server supports these characters, but they are illegal characters in XML. For example, the table name "@Table1" becomes "_x0040_Table1" in the generated schema. These characters will appear in generated schemas—do not delete or change them.

- BizTalk Server does not support using spaces in SQL table names used by the SQL adapter. If you have existing tables whose names contain spaces, you should use an alias for the table name. For more information about creating aliases, see the Microsoft Visual Database Tools topic "Creating Table Aliases" at http://go.microsoft.com/fwlink/?LinkID=24927.

SQL Adapter Security Recommendations

You use the SQL adapter to exchange data between BizTalk Server and a SQL Server database. For more information about the SQL adapter,

The SQL adapter does not run SQL commands within a message, so it is not vulnerable to injection attacks. The SQL adapter only runs stored procedures and updategrams.

It is recommended you follow these guidelines for securing and deploying the SQL adapter in your environment.

- Depending upon your scenario, your SQL Server database that the SQL Server adapter uses may be remote (if retrofitting an existing application) or within the BizTalk servers (new application).

- You should use a separate database for this adapter than the databases you use for BizTalk Server.

- It is recommended that you use Windows Authentication to connect to SQL Server. If you use SQL Server security, BizTalk Server stores the connection string in encrypted form in the Credential database.

- If you use Windows Server 2003, you must manually enable remote data transaction coordinator (DTC) because the SQL adapter runs distributed transactions.

Windows SharePoint Services Adapter

The BizTalk Server 2006 adapter for Microsoft Windows SharePoint Services provides a tighter integration of BizTalk Server with Windows SharePoint Services and Microsoft Office. Using the Windows SharePoint Services adapter in your BizTalk Server 2006 solution provides you with the following capabilities:

- Easy access to input and output messages through Windows SharePoint Services.
- The ability to edit XML messages by using Office applications such as Microsoft Office InfoPath 2003.
- Two-way transformations of XML messages to and from InfoPath.

In This Section

- What Is the Windows SharePoint Services Adapter?
- Setting Up and Deploying the Windows SharePoint Services Adapter
- Configuring the Windows SharePoint Services Adapter
- Windows SharePoint Services Adapter Walkthroughs

What Is the Windows SharePoint Services Adapter?

The BizTalk Server 2006 adapter for Windows SharePoint Services provides a tighter integration with Windows SharePoint Services and Microsoft Office InfoPath. The following topic describes the features and an overview of how the Windows SharePoint Services adapter works.

Features of the Windows SharePoint Services adapter

The following list describes important features of the Windows SharePoint Services adapter:

- The ability to send BizTalk Server XML and binary messages to SharePoint document libraries.
- Integration with InfoPath: You can transform outgoing BizTalk Server XML messages to automatically open in InfoPath when opened from the Windows SharePoint Services site.
- Property promotion for messages going into Windows SharePoint Services. Up to 16 SharePoint columns can be updated with BizTalk Server metadata about the message-like orchestration instance ID, message ID, or values extracted from the message.

- File-name definition based on message content and BizTalk Server properties.

- The ability to send documents to an arbitrary list (instead of to a document library): In this case the document itself is not stored in Windows SharePoint Services but the property promotion still happens so a new list item is created and the column values are retrieved from the message.

- The ability to receive messages from any view of any document library and archive them to a specified document library using the specified file name.

- Promotion of Windows SharePoint Services adapter properties in BizTalk Server: Windows SharePoint Services file information is made available in BizTalk Server as message context properties. The message context properties can be accessed from pipelines, orchestrations, etc. Custom SharePoint columns can be accessed through the WSS.InPropertiesXml document.

- Full support for dynamic ports: Send adapters can support static URI binding (defined by the user when the send port is created) or dynamic URI binding (defined by the orchestration when sending the message). All configuration information can be defined through message context properties, such as WSS.Filename and WSS.ConfigTimeout, for dynamic send ports as well as physical send ports.

- Performance counters

How the Windows SharePoint Services adapter works

The BizTalk Server 2006 adapter for Windows SharePoint Services consists of three main components:

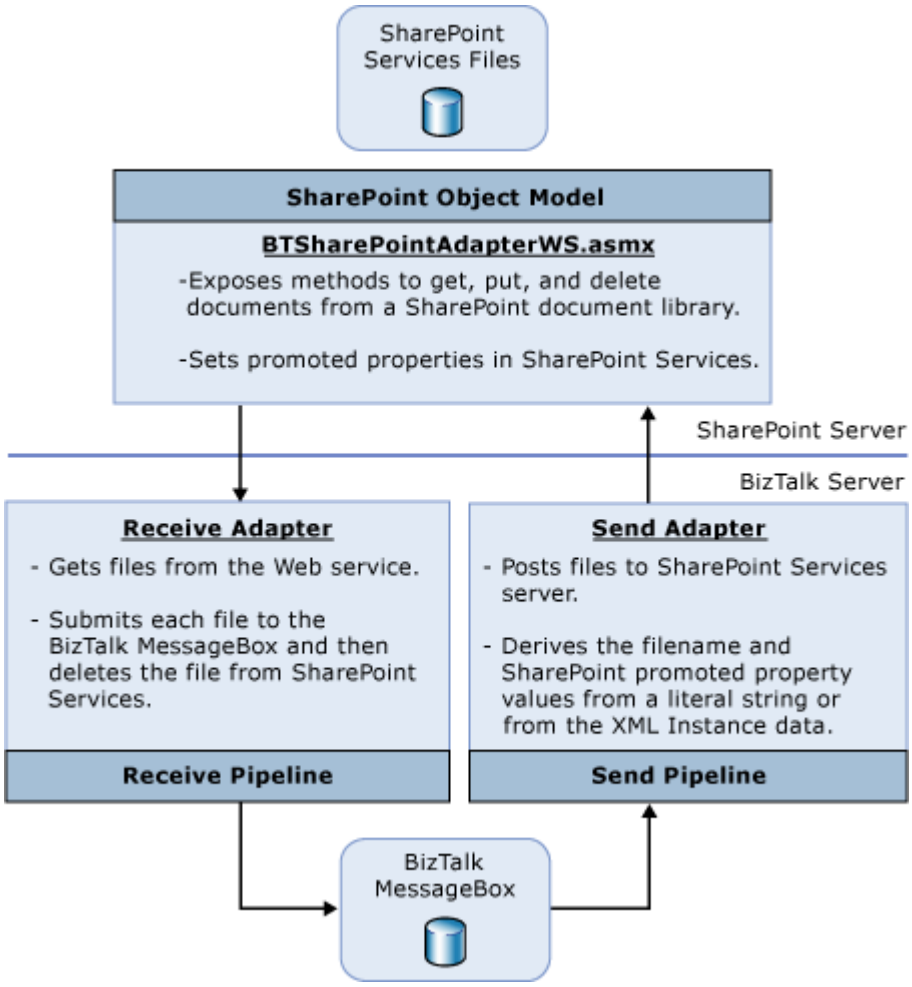
- Windows SharePoint Services adapter Web service

- Windows SharePoint Services receive adapter

- Windows SharePoint Services send adapter

On the Windows SharePoint Services server, the Web service (BTSharePointAdapterWS.asmx) is installed to provide access to the Windows SharePoint Services libraries and lists. The Web service exposes methods to get, put, delete, and archive documents from a SharePoint library. The receive adapter retrieves files from the Web service and the send adapter posts files to it.

The following figure shows the main components of the BizTalk Server 2006 adapter for Windows SharePoint Services that provide these capabilities.



Receiving documents from Windows SharePoint Services

The receive adapter polls Windows SharePoint Services document library views. It calls a Web method on the Windows SharePoint Services server which uses the Windows SharePoint Services object model to browse the library, check out the files and return the file data to the adapter. The adapter then submits the files to the BizTalk Server MessageBox and calls another Web method to delete or archive the files from Windows SharePoint Services. In order to filter files in a Windows SharePoint Services library, the adapter polls the Windows SharePoint Services library through a Windows SharePoint Services view.

The centralized (polling) approach offers a simple management model where configuration is done on the BizTalk server. It also offers better performance due to the fact that it allows batching of the messages.

Since platform-level transaction support is not available across Windows SharePoint Services, Web services, and BizTalk Server, the check-out mechanism is used to minimize errors associated with failure conditions. Under certain conditions (that is, files are successfully sent into the BizTalk Server MessageBox database but cannot be deleted from Windows SharePoint Services), the files will remain checked out on the Windows SharePoint Services server even

though they were submitted to BizTalk Server. Errors will be logged to the event log on the BizTalk server.

Sending documents to Windows SharePoint Services

The adapter sends documents to Windows SharePoint Services by calling a Web method on the Windows SharePoint Services server. The adapter specifies the Windows SharePoint Services site URL, document library or list URL relative to the site, file, or list item name and promoted properties to associate with the file.

You can set the file name to a fixed string or to a name derived from the XML data in the document. Deriving the name can be very useful to enforce standard naming conventions. The adapter can also set promoted property values on the file as column values. As with the file names, the promoted property values can be fixed or can be derived from the XML data in the document.

Windows SharePoint Services promoted properties are used to make XML elements visible when browsing a Windows SharePoint Services forms library. When an InfoPath form is published to a Windows SharePoint Services forms library, InfoPath configures the forms library to promote key elements, making this happen automatically. This feature is available in Windows SharePoint Services only when using InfoPath form libraries (document libraries that store InfoPath forms with the same XSD schema and InfoPath solution).

Windows SharePoint Services adapter property promotion enables the user to promote properties into Windows SharePoint Services when documents with different schemas are stored in the same document library.

BizTalk Server property promotion is a similar concept, only that properties are made visible to the orchestration as properties on the message and not to the end user on the UI. In addition, BizTalk Server supports a concept of property demotion when the property values are saved back into the document.

When using the Windows SharePoint Services adapter with InfoPath forms and forms libraries (rather than arbitrary XML and document libraries), you do not need to set the promoted properties through the send adapter. Instead, the document can be changed within the orchestration (directly by changing the message or indirectly through properties that will be demoted). The values will be automatically promoted by Windows SharePoint Services.

Security considerations for the Windows SharePoint Services adapter

The Windows SharePoint Services adapter consists of subsystems, the BTSharePointAdapterWS Web service that runs on the Windows SharePoint Services Web site, and the adapter runtime that runs on the BizTalk server within the BizTalk Server host instance process. The adapter runtime invokes the BTSharePointAdapterWS Web service which must have permissions to perform certain tasks within Windows SharePoint Services. Since this component runs as the caller, the permissions need to be granted to the caller. This means that the BizTalk host instance must be made a **Contributor** on the SharePoint site in order to be able to send and receive messages from that site. The BTSharePointAdapterWS Web service can be invoked only by members of the **SharePoint Enabled Hosts** group. In order to allow a BizTalk host instance, running the adapter runtime, to interact with the Web

service, the host instance Windows account must be made a member of the **SharePoint Enabled Hosts** group. It is the responsibility of the administrator to add and remove accounts from this group as well as to make the host instance accounts members of the SharePoint **Contributor** role.

Component	Process identity	Permission
BTSharePointAdapterWS Web service	Caller identity	Invoke permission granted to SharePoint Enabled Hosts group
Adapter runtime	Identity of BizTalk host	N/A
Windows SharePoint Services Object Model	N/A	The SharePoint Enabled Hosts group must be a member of the **Contributor** role in SharePoint Services.

BizTalk Server Setup configures the permissions on the BTSharePointAdapterWS Web service so that only the accounts that are members of the **SharePoint Enabled Hosts** group can access this Web service. If you want hosts to run the Windows SharePoint Services adapter, the administrator will have to add the NT group associated with that host to the **SharePoint Enabled Hosts** group and also add the **SharePoint Enabled Hosts** group to the Windows SharePoint Services **Contributor** role.

Permissions to Windows SharePoint Services files, lists, and document libraries are restricted using Windows SharePoint Services security. The messages are sent from Windows SharePoint Services directly into BizTalk Server. The communication between the adapter runtime and the Web service is done over HTTP or HTTPS.

The adapter assumes that the BTSharePointAdapterWS Web service is using the same HTTP scheme (HTTP or HTTPS) as the Windows SharePoint Services site. This means that the adapter will use HTTPS to communicate with the BTSSharePointAdapterWS Web service when the Windows SharePoint Services Site is created on a secure IIS Web site, or it will use HTTP to communicate with the BTSharePointAdapterWS Web service when the Windows SharePoint Services site is created on an IIS Web site without a server certificate.

Setting Up and Deploying the Windows SharePoint Services Adapter

The topics in this section discuss setting up and deploying the Windows SharePoint Services adapter in both a single-server deployment and a multiserver deployment.

In This Section

- Single-Server Deployment
- Multiserver Deployment

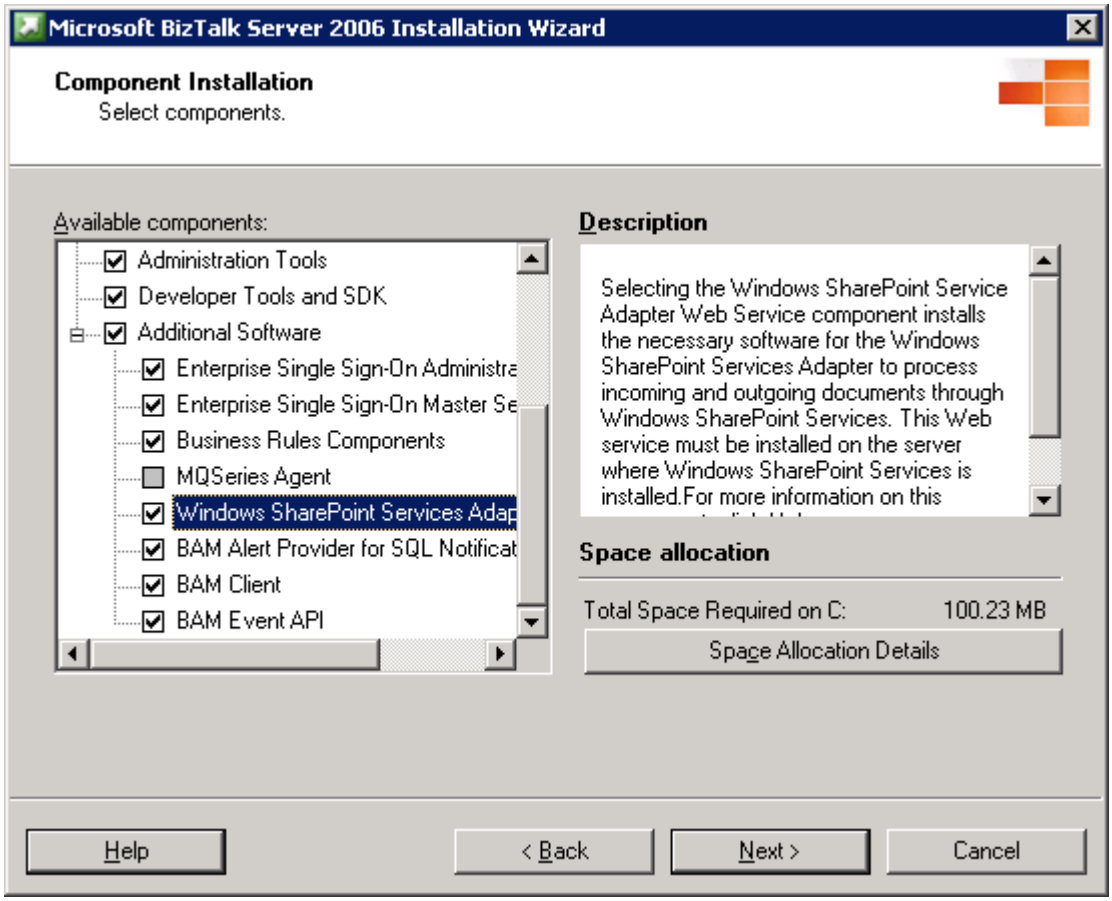
Single-Server Deployment

This topic discusses single-server setup and deployment considerations for the BizTalk Server 2006 adapter for Windows SharePoint Services.

Installing the Windows SharePoint Services adapter in a single-server deployment

Selecting the Windows SharePoint Service adapter Web service component installs the necessary software for the Windows SharePoint Services adapter to process incoming and outgoing documents through Windows SharePoint Services. This Web service must be installed on the server where Windows SharePoint Services is installed. The adapter Web service can handle multiple SharePoint sites including the Web site that hosts Business Activity Services (BAS), regardless of whether they are on the same IIS site or on different IIS sites.

The following screen shows where you make the selection to install the Windows SharePoint Services adapter.



The Windows SharePoint Services adapter has three components:

- Runtime components

- Design time components
- Adapter Web service

The adapter runtime is installed and configured automatically by the BizTalk Server Runtime feature. The adapter design time components are installed and configured with the other BizTalk Server features. You interact with the design time components by creating Windows SharePoint Services ports through tools that are included in the Administration Tools, Developer Tools, and SDK or BizTalk Server Runtime features. You cannot customize any configuration options for runtime and design time components. You can customize only the Windows SharePoint Service adapter Web service options.

Only members of the SharePoint Enabled Hosts group have permissions to invoke the adapter Web service. For more information about the Windows SharePoint Services permissions needed by the Windows SharePoint Services adapter runtime,.

To install the Windows SharePoint Services adapter

1. Install BizTalk Server 2006.
2. On the **Component Installation** screen, under **Available Components**, under **Additional Software**, select **Windows SharePoint Services Adapter Web service**.

Configuring the Windows SharePoint Services adapter Web service in a single-server deployment

You can configure the Windows SharePoint Services adapter using either a basic configuration or a custom configuration. For more information about these tools,

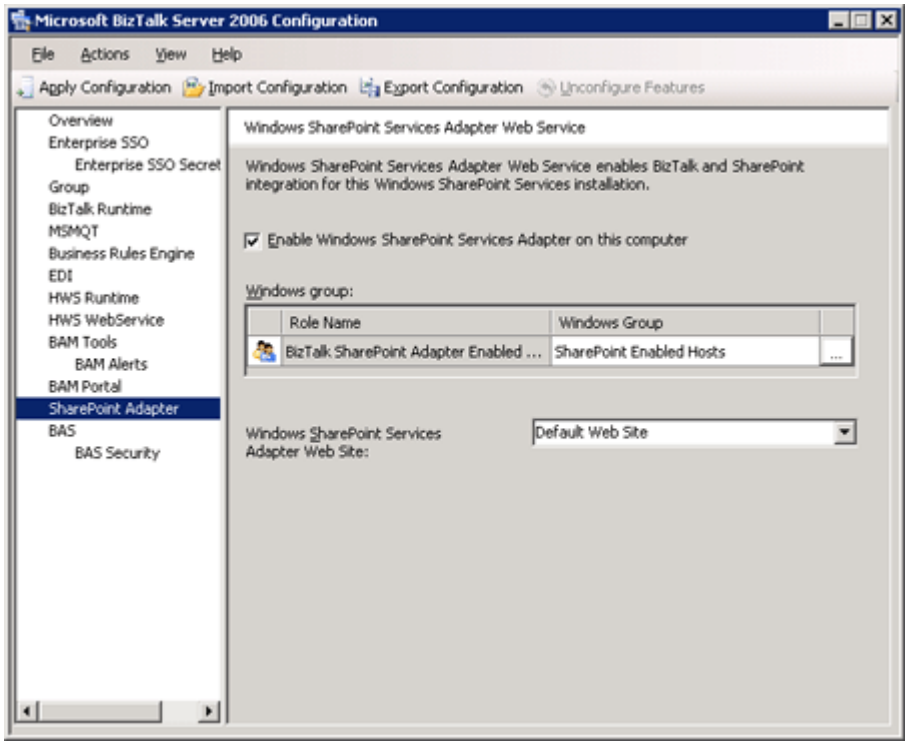
Using a basic configuration

BizTalk Server 2006 allows you to configure the server by using default settings. The default settings for the server are configured using the database server name, user name, and password that you enter into the configuration wizard. When you configure the Windows SharePoint Services adapter Web service using a basic configuration, the following happens:

- The SharePoint Enabled Hosts Windows group is created.
- The Default Web Site is used to host the Windows SharePoint Services Adapter
- The BTSSharePointAdapterWSAppPool application pool is created and configured to run under the account that is also used to run the Windows SharePoint Services application pool.
- The BTSharePointAdapterWS virtual application is created and configured to run in the BTSSharePointAdapterWSAppPool application pool
- The BTSharePointAdapterWS virtual application contains the Web service

Using a custom configuration

The custom configuration manager provides a high-level analysis of the configuration state of the features you have installed on the local computer. The tool allows you to configure and unconfigure features, configure security settings, and import and export configurations from other computers.



Use the **Windows SharePoint Services Adapter Web Service** page to configure the Windows SharePoint Services adapter on this computer. The following table lists the configuration options.

Use this	To do this
Enable Windows SharePoint Services Adapter on this computer	Select **Enable Windows SharePoint Services Adapter on this computer** to enable the adapter on this computer.
Windows group	The **Windows group** list provides a view that you can edit of the BizTalk SharePoint Adapter Enabled Hosts Windows group.
Windows SharePoint Services Adapter Web site	Select the Web site that will host the Windows SharePoint Service adapter Web service.

When you configure the Windows SharePoint Services adapter using a custom configuration, the following happens:

- The SharePoint Enabled Hosts Windows group is created by default unless you specify another Windows group
- The Default Web Site is used to host the Windows SharePoint Services adapter unless you specify another Web site
- The BTSSharePointAdapterWSAppPool application pool is created and configured to run under the account that is also used to run the Windows SharePoint Services application pool
- The BTSharePointAdapterWS virtual application is created and configured to run in the BTSSharePointAdapterWSAppPool application pool
- The BTSharePointAdapterWS virtual application contains the Web service

To configure the Windows SharePoint Services adapter by using a custom configuration

1. In the **custom configuration manager**, select the **SharePoint adapter** node.
2. Select **Enable Windows SharePoint Services Adapter on this computer**.
3. Under **Windows Group**, select the Windows group you will be using for the Windows SharePoint Services adapter. By default, this is SharePoint Enabled Hosts.
4. In the **Windows SharePoint Services Adapter Web Site** drop-down box, select the Web site where the adapter components will be installed. By default, this is the Default Web Site.
5. Click **Apply Configuration**.

Considerations for a single-server deployment

When you set up and deploy the Windows SharePoint Services adapter in a single-server environment, consider the following:

- Add the BizTalk Service account to the SharePoint Enabled Hosts Windows group on that server.
- Add the SharePoint Enabled Hosts group to the SharePoint Contributors role using the SharePoint Central Administration tool.
- The Web site that you install the Web service on must be extended as a SharePoint Services Web site.
- Windows SharePoint Services 2.0 with Service Pack 2 is required.
- You can install and configure the Windows SharePoint Services adapter using silent installation.

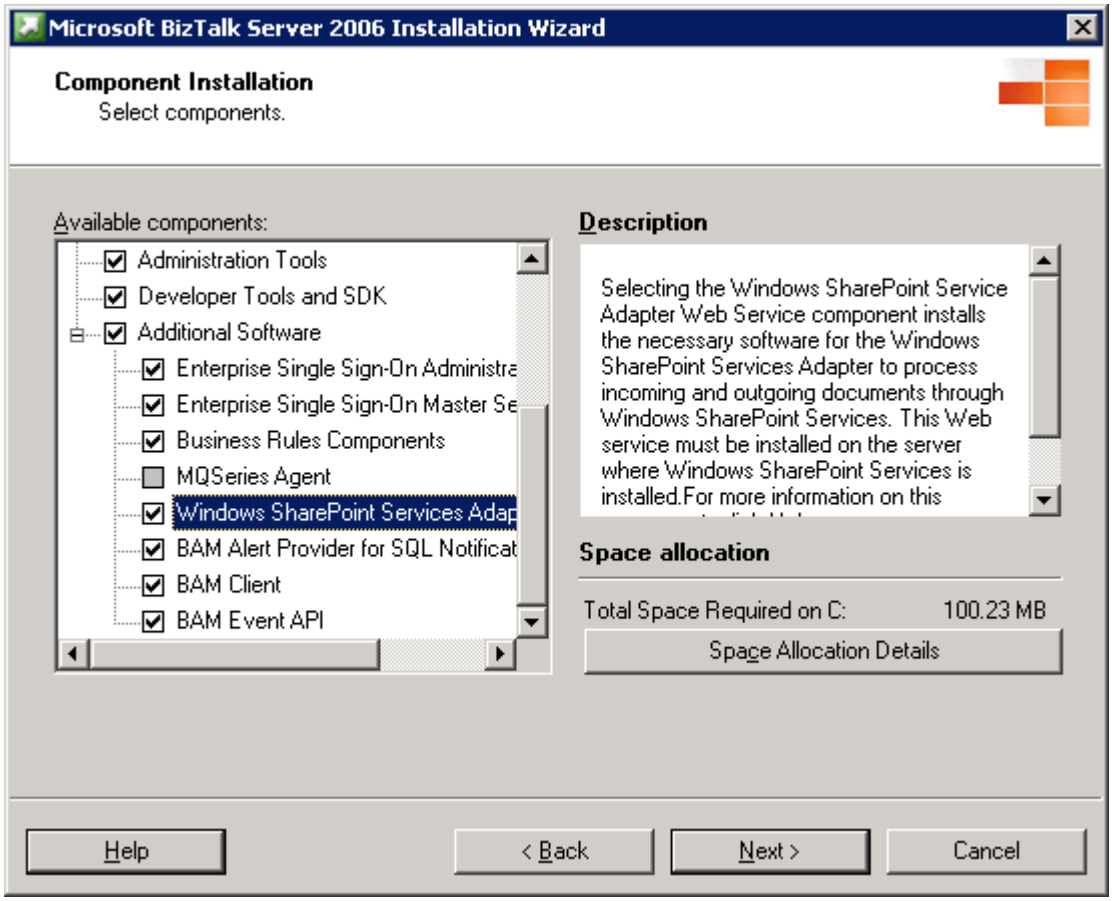
Multiserver Deployment

This topic discusses multiserver setup and deployment considerations for the BizTalk Server 2006 adapter for Windows SharePoint Services.

Installing the Windows SharePoint Services adapter in a multiserver deployment

Selecting the Windows SharePoint Service Adapter Web Service component installs the necessary software for the Windows SharePoint Services adapter to process incoming and outgoing documents through Windows SharePoint Services. This Web service must be installed on the server where Windows SharePoint Services is installed. The adapter Web service can handle multiple SharePoint sites including the Web site that hosts Business Activity Services (BAS) regardless of whether they are on the same IIS site or on different IIS sites.

The following screen shows where you make the selection to install the Windows SharePoint Services adapter.



The Windows SharePoint Services Adapter has three components:

- Runtime components

- Design time components
- Adapter Web service

The adapter runtime is installed and configured automatically by the BizTalk Server Runtime feature. The adapter design time components are installed and configured with other BizTalk Server features. You interact with the design time components by creating Windows SharePoint Services ports through tools that are included in the Administration Tools, Developer Tools, and SDK or BizTalk Server Runtime features. You cannot customize any configuration options for runtime and design time components. You can customize only the Windows SharePoint Services adapter Web Service options.

Only members of the SharePoint Enabled Hosts group will have permissions to invoke the adapter Web service. For more information about the Windows SharePoint Services permissions needed by the Windows SharePoint Services adapter runtime,.

To install the Windows SharePoint Services adapter

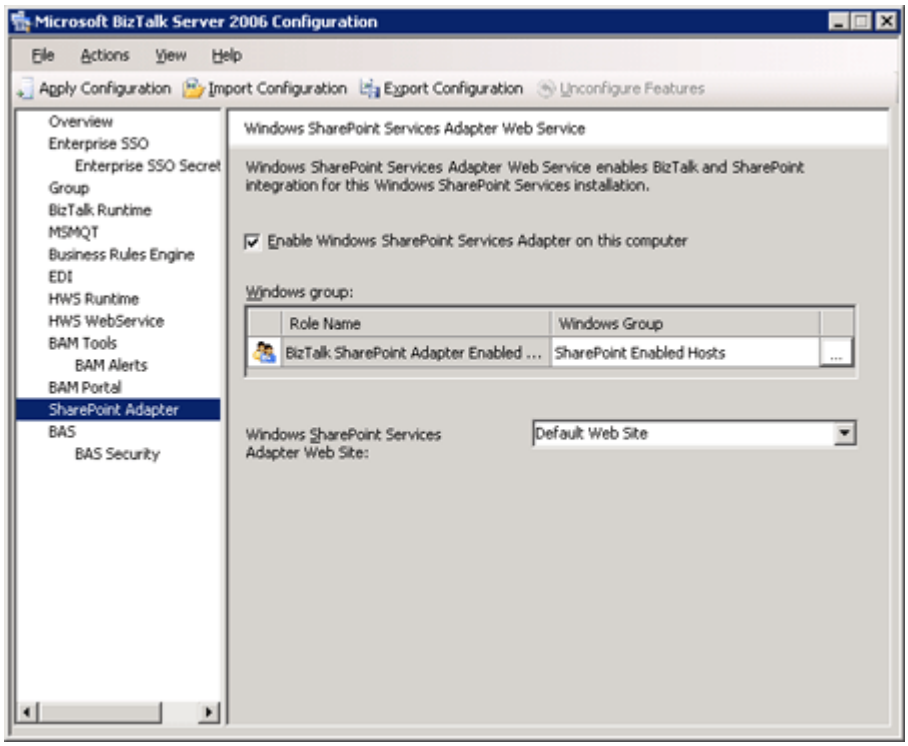
1. Install BizTalk Server 2006.
2. On the **Component Installation** screen, under **Available Components**, under **Additional Software**, select **Windows SharePoint Services Adapter Web service**.

Configuring the Windows SharePoint Services adapter Web service in a multiserver deployment

You configure the Windows SharePoint Services adapter using the custom configuration manager.

Using a custom configuration

The custom configuration manager provides a high-level analysis of the configuration state of the features you have installed on the local computer. The tool allows you to configure and unconfigure features, configure security settings, and import and export configurations from other computers.



Use the **Windows SharePoint Services** page to configure the Windows SharePoint Services adapter on this computer. The following table lists the configuration options.

Use this	To do this
Enable Windows SharePoint Services Adapter on this computer	Select **Enable Windows SharePoint Services Adapter on this computer** to enable the adapter on this computer.
Windows group	The **Windows group** list provides a view that you can edit of the BizTalk SharePoint Adapter Enabled Hosts Windows group.
Windows SharePoint Services Adapter Web site	Select the Web site that will host the Windows SharePoint Services adapter Web service.

When you configure the Windows SharePoint Services adapter using custom configuration, the following happens:

- The SharePoint Enabled Hosts Windows group is created by default unless you specify another Windows group
- The Default Web Site is used to host the Windows SharePoint Services adapter unless you specify another Web site

- The BTSSharePointAdapterWSAppPool application pool is created and configured to run under the account that is also used to run the Windows SharePoint Services application pool

- The BTSharePointAdapterWS virtual application is created and configured to run in the BTSSharePointAdapterWSAppPool application pool

- The BTSharePointAdapterWS virtual application contains the Web service

To configure the Windows SharePoint Services adapter by using custom configuration

1. In the **custom configuration manager**, select the **SharePoint adapter** node.

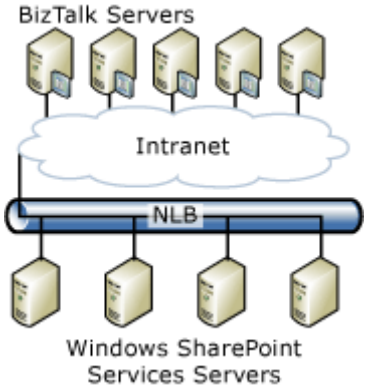
2. Select **Enable Windows SharePoint Services Adapter on this computer**.

3. Under **Windows Group**, select the Windows group you will be using for the Windows SharePoint Services adapter. By default, this is SharePoint Enabled Hosts.

4. In the **Windows SharePoint Services Adapter Web Site** drop-down box, select the Web site where the adapter components will be installed. By default, this is the Default Web Site.

5. Click **Apply Configuration**.

Considerations for a multiserver deployment



General considerations

When you set up and deploy the Windows SharePoint Services adapter in a multiserver environment, consider the following:

- Add the BizTalk Service account to the SharePoint Enabled Hosts Windows group on each server.

- Add the SharePoint Enabled Hosts group to the SharePoint Contributors role using the SharePoint Central Administration tool.

- The Web site that you install the Web service on must be extended as a SharePoint Services Web site.
- Windows SharePoint Services 2.0 with Service Pack 2 is required.
- You can install and configure the Windows SharePoint Services adapter using a silent installation.
- Considerations for network load balancing (NLB)

The BizTalk Server 2006 adapter for Windows SharePoint Services supports NLB clustering of the Windows SharePoint Services servers along with multiple BizTalk servers that are configured in the same group. For this, Windows SharePoint Services must be installed on the NLB cluster as recommended by SharePoint documentation.

When you set up and deploy the Windows SharePoint Services adapter in a multiserver environment with NLB, consider the following:

- Configure Windows SharePoint Services by selecting the option to point to an existing BizTalk Management database. Point to the BizTalk Management database created on the first computer. This indicates to Windows SharePoint Services that you intend to have a Web server environment. Extend the Web site by pointing to the existing content database.
- You must extend the same Web site (for example, the default Web site on port 80) on each Windows SharePoint Services computer in the Web server environment. For more information about Windows SharePoint Services Web server environments,
- The BTSharePointAdapterWS must be installed and configured the same way on each of the NLB hosts. You must configure the following NLB settings:
  - Protocol: TCP
  - Ports: 80 (The HTTP Port of the IIS Web site where the Windows SharePoint Services adapter Web service has been installed and configured)
  - Filtering mode: Multiple host
  - Affinity: None

Configuring the Windows SharePoint Services Adapter

The topics in this section describe how to configure the Windows SharePoint Services adapter.

In This Section

- How to Configure a Windows SharePoint Services Receive Location
- How to Configure a Windows SharePoint Services Send Handler

- How to Configure a Windows SharePoint Services Send Port
- How to Configure Send Ports Using Windows Sharepoint Services Context Properties
- Windows SharePoint Services Adapter Properties Reference
- Windows SharePoint Services Adapter Expressions
- Supported Windows SharePoint Services Column Types

How to Configure a Windows SharePoint Services Receive Location

This topic describes how to create and configure a Windows SharePoint Services receive location by using the BizTalk Server Administration Console.

To create and configure a Windows SharePoint Services receive location

1. Ensure you have a receive port properly configured.
2. In the **BizTalk Server Administration console**, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group [GroupName]**, expand **Applications**, and then expand the application you want to create a receive location in.
3. Right-click **Receive Locations**, click **New**, and then click **One-way Receive Location**.
4. Select the receive port that will contain this receive location, and then click **OK**.
5. In the **Receive Location Properties** dialog box, under **Transport**, in the **Type** drop-down box, select **Windows SharePoint Services**.
6. Click **Configure**.
7. In the **Windows SharePoint Services Transport Properties** dialog box, do the following:

Use this	To do this
Adapter Web Service Port	The HTTP port of the IIS Web site where the Windows SharePoint Services adapter Web service is installed. By default, this is the Default Web Site configured on port 80. If you have configured the Windows SharePoint Services Web service on any IIS Web site other than the Default Web Site, you will have to update this value.
Timeout	The time-out, in milliseconds, for the adapter runtime Web service calls made to the Windows SharePoint Services adapter Web service. You may need to increase this value if the message or batch size is higher than the average that is expected by the adapter.

Archive Filename	(Optional) The archived file Windows SharePoint Services file name. You can type in a literal value like 'PurchaseOrder0001.xml' or an expression. Expressions can include any mix of literals, macros, and XPATH queries. For example, "PurchOrd-%XPATH=//po:PurchaseOrderId%-%MessageID%.xml". When no file name is supplied the file name of source file is used..
Archive Location URL	The Windows SharePoint Services folder URL, relative to the SharePoint site, where the processed files are archived. For example, Archive or /Shared Documents/Processed Orders/. If an archive location is not specified, the document is deleted after being processed by the adapter.
Archive Overwrite	Determines whether existing files in the archive are overwritten. Select "Yes" to overwrite existing files. Select "No" for the archive to fail if a file with the same name already exists in the archive. In this case the file will remain checked out and it will have to be archived manually.
Batch Size	The maximum number of documents that the Windows SharePoint Services Messaging Adapter Web service will process as a batch. A processed batch might contain fewer messages than the defined batch size; however, it will never contain more messages.
Error Threshold	The maximum number of consecutive polling failures encountered by the adapter until the receive location is disabled. Set this field to 0 in order to never disable the receive location.
Namespace Aliases	(Optional) A comma or semicolon-delimited list of namespace aliases definitions. Use this field to define the namespace aliases that are used by the XPATH queries introduced in the Archive Filename field. For example, po='http://OrderProcess/POrder', conf='http://OrderProcess/Confirmation' ipsol='{D8217CF1-4EF7-4bb5-A30D-765ECB09E0D9}'
Polling Interval	The time interval, in seconds, between two consecutive queries performed by the adapter to see if any new messages are available for processing.
SharePoint Site URL	The complete URL of the Windows SharePoint Services Web site. For example, http://BizTalkServer/sites/BASSite. You may specify any Windows SharePoint Services site including the site created by the BizTalk Server Business Activity Services (BAS) feature.
Source Document Library URL	This is the URL of the Windows SharePoint Services document library, relative to the SharePoint site, where the documents are retrieved. For example, /Shared Documents/ or /New Purchase Orders/.

View Name	This is the Windows SharePoint Services view used to filter documents processed by the adapter. For example, Approved Orders. Leave this field empty to process all the existing documents in the source document library. Folders showing up in a view and the messages contained in those folders will not be processed by the adapter. You can create flat views that will show all documents in a flat structure including documents existing in subfolders.
Microsoft Office Integration	"Optional" to attempt to remove InfoPath processing instructions if possible or to process as is if not possible (for instance, a binary document). Choose "Yes" to remove InfoPath processing instructions or to skip the message in case of an error. Choose "No" to process the document "as is." For binary messages, you must use “No” or “Optional” values.

8. Click **OK**.

How to Configure a Windows SharePoint Services Send Handler

Use the following procedure to change the host with which the Windows SharePoint Services send handler is associated.

Procedures

To change global variables for a Windows SharePoint Services send handler

1. In the BizTalk Server Administration Console, click to expand **BizTalk Server 2006 Administration**, and then click to expand **BizTalk Group [<servername>:<management database>]**, click to expand **Platform Settings**, and then click to expand **Adapters**. The list of adapters appears under the folder.

2. Click **Windows SharePoint Services**, and in the right pane, right-click the send handler that you want to configure, and then click **Properties**.

3. In the **Adapter Handler Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the send handler will be associated.

4. On the **General** tab, click **Properties**.

5. In the **Windows SharePoint Services Transport Properties** dialog box, do the following:

Use this	To do this
Send Batch Size	The maximum number of documents that the Windows SharePoint Services Web service will process as a batch. The default is 20.

How to Configure a Windows SharePoint Services Send Port

This topic describes how to create and configure a Windows SharePoint Services send port by using the BizTalk Server Administration Console.

To create and configure a Windows SharePoint Services send port

1. In the BizTalk Server Administration Console, create a new send port or double-click an existing send port to modify it. Configure all of the send port options and specify **Windows SharePoint Services Transport Properties** for the **Type** option in the **Transport** section of the **General** tab.

2. On the **General** tab, in the **Transport** section, next to **Type**, click **Configure**.

3. In the **Windows SharePoint Services Transport Properties** dialog box, do the following:

Use this	To do this
Adapter Web Service Port	The HTTP port of the IIS Web site where the Windows SharePoint Services adapter Web service is installed. By default, this is the Default Web Site configured on port 80. If you have configured the Windows SharePoint Services Web service on any other IIS Web site than the Default Web Site, you will have to update this value.
Timeout	The time-out, in milliseconds, for the adapter runtime Web service calls made to the Windows SharePoint Services adapter Web service. You may need to increase this value if the message or batch size is higher than the average that is expected by the adapter.
Destination Folder URL	The Windows SharePoint Services destination folder URL, relative to the SharePoint site. For example, Shared Documents, Shared Documents/Purchase Orders/, or Lists/Tasks. You can send messages to a SharePoint list by specifying the URL of the list, for example, Lists/Tasks. If you specify a list as a destination, the message body will not be saved with the list item but the values extracted from the message will still be promoted into the SharePoint columns.
Filename	(Optional) The Windows SharePoint Services filename. You can type in a literal value like 'PurchaseOrder0001.xml' or an expression. Expressions can include any mix of literals, macros and XPATH queries, ex: "PurchOrd-%XPATH=//po:PurchaseOrderId%-%MessageID%.xml". When no file name is supplied, the file name will be the name of the original file, the value supplied by the orchestration, or 'Msg-%MessageID%.xml' if the orchestration does not define the file name.
Namespaces	(Optional) A comma or semicolon-delimited list of namespace aliases definitions. Use this field to define the namespace aliases that are used by

Aliases	the XPATH queries introduced in fields like 'Filename' or 'Column Value'. For example, po='http://OrderProcess/POrder', conf='http://OrderProcess/Confirmation' xmlns=""; ipsol='{D8217CF1-4EF7-4bb5-A30D-765ECB09E0D9}'.
Overwrite	Determines whether an existing file is overwritten. Select 'Yes' to overwrite existing files. Select 'No' to raise an error and suspend the message when a file with the same name already exists. Select 'Rename' to rename the file. Select 'Orchestration' to use the value defined by the orchestration.
SharePoint Site URL	The complete URL of the Windows SharePoint Services Web site. For example, http://BizTalkServer/sites/BASSite. You may specify any Windows SharePoint Services site including the site created by the BizTalk Server Business Activity Services (BAS) feature.
Microsoft Office Integration	'Optional' to change the document so that it automatically opens in an Office application like InfoPath or to save the document as-is if no InfoPath solution is found. 'Yes' to change the document so that it automatically opens in an Office application like InfoPath or suspend the message if no InfoPath solution is found. 'Yes (InfoPath Form Library)' to change the document to automatically open in an Office application such as InfoPath using the InfoPath solution found in the form library if the message is sent to a Windows SharePoint Services InfoPath form library. If the form library does not have an InfoPath solution, the message will be suspended. 'No' to save the document 'as-is' without any changes. 'Orchestration' to use the value defined by the orchestration. For binary messages, No or Optional values must be used.
Templates Document Library	Enter the name of a SharePoint document library where the InfoPath solutions are stored. For example, *My Solutions*. This is the first place where the adapter will look for a matching InfoPath solution. If a solution is not found, the adapter will look in the Templates fallback document library.
Templates Fallback Document Library	Enter the name of a SharePoint document library where the InfoPath solutions are stored. For example, Templates. The adapter will only search this document library for a matching InfoPath solution if a solution is not found in the templates document library. The 'Templates Fallback Document Library' and 'Templates Document Library' fields can be used with two sets of InfoPath solutions. There are generic InfoPath solutions that work for all general purposes, and specialized InfoPath solutions that are used only for a particular partner. The 'Templates Fallback Document Library' field should point to the generic solutions, and the 'Templates Document Library' should point to the specialized solutions for that particular partner.

Templates Fallback Namespace Column	This is the name of the Templates Fallback Document Library SharePoint column that stores the namespace of the InfoPath solution. For example, Namespace.
Templates Namespace Column	This is the name of the Templates Document Library SharePoint column that stores the namespace of the InfoPath solution. For example, Namespace.
Column *n*	This is the name of the Windows SharePoint Services column that exists in the destination document library. This is the column that should be updated with the value extracted from the message or specified in the 'Column value' field.
Column *n* Value	Enter the column value to be set for this message. You can type in a literal value like 'Purchase Order' or an expression. Expressions can include any mix of literals, macros, and XPATH queries. For example, "%XPATH=//po:POAmount%", "%SendingOrchestrationID%".

4. Click **OK** and **OK** again to save settings.

How to Configure Send Ports Using Windows Sharepoint Services Context Properties

This topic describes how to configure Windows SharePoint Services send ports at runtime using Windows Sharepoint Services context properties from a BizTalk orchestration. The same mechanism can be used to configure Windows SharePoint Services dynamic and late-bound send ports.

To set configuration properties for a send port using Windows Sharepoint Services adapter context properties

The configuration properties for a dynamic send port are set in an orchestration at runtime. Adapter properties that are exposed in the **Windows SharePoint Services Transport Properties** dialog box can also be applied to a dynamic or late bound send port. To set configuration properties for a dynamic or late bound send port using the Windows Sharepoint Services adapter context properties follow these steps:

1. For dynamic send ports, follow the steps in the topic How to Create a Send Port to create a Dynamic One-way send port.

2. Use a **Message Assignment** shape within a **Construct Message** shape in an orchestration to set the configuration properties for the outbound message. For an example of how to set the configuration properties for an outbound message see

Walkthrough: Module 3 - Accessing SharePoint Properties from an Orchestration . The **Construct a new message** section of this topic illustrates how to set configuration properties of an outbound message. The adapter context properties that correlate to the properties that can be set in the **Windows SharePoint Services Transport Properties** dialog box are listed in the table below:

Transport Property	**Adapter Context Property**	**Data Type**	**Comments**
Adapter Web Service Port	WSS.ConfigAdapterWSPort	Int	Valid values are from 1 to 65535 The default value is 80
Timeout	WSS.ConfigTimeout	Int	Valid values are from 1000 to 2147483647 The default value is 100000 Specify a value of 0 to indicate an infinite timeout.
Destination Folder URL	NA	NA	For dynamic ports, this is set indirectly by setting t **Microsoft.XLANGs.BaseTypes.Address** property of t dynamic port with an expression shape in an orchestration. F late-bound ports this property cannot be set at runtime since it always overridden by the physical send port value.
Filename	WSS.Filename	String	Supports the use of all filename macros that can be used in t transport properties except for the **%Filename%** a **%Extension%** macros.
Namespaces Aliases	WSS.ConfigNamespaceAliases	String	If a namespace alias set for a message at runtime exac matches the namespace alias set for the send port that t message is routed to then the namespaces are merged and routing error occurs. To prevent this problem ensure that t specified namespace aliases are not identical. For example if t following expression is used in an orchestration to set t namespace alias for a message: and if this message is routed to a send port that specifies t following value for the **Namespace Aliases** property: orchns='http://OrderProcess.PurchaseOrder' then an error will occur when BizTalk Server attempts to route t message to this send port. To resolve this problem you cou specify the following value for the **Namespace Aliases** proper of the send port: **orchns2**='http://OrderProcess.PurchaseOrder'

Overwrite	WSS.ConfigOverwrite	String	Valid values are: • "yes" • "no" • "rename"
SharePoint Site URL	WSS.InListUrl	String	For dynamic ports, this is set indirectly by setting t **Microsoft.XLANGs.BaseTypes.Address** property of t dynamic port with an expression shape in an orchestration. F late-bound ports this property cannot be set at runtime since it always overridden by the physical send port value.
Microsoft Office Integration	WSS.ConfigOfficeIntegration	String	Valid values are: • "yes" • "no" • "yesformlibrary" • "optional"
Templates Document Library	WSS.CustomConfigTemplatesDocLib	String	None
Templates Fallback Document Library	WSS.ConfigCustomTemplatesDocLib	String	None
Templates Fallback Namespace Column	WSS.ConfigCustomTemplatesNamespaceCol	String	None
Templates Namespace Column	WSS.ConfigTemplatesNamespaceCol	String	None
Column *n*	WSS.ConfigPropertiesXml Column name is set in <PropertyName*x*>*columnname*</	String	None

	PropertyName*x*> field.		
Column *n* Value	WSS.ConfigPropertiesXml Column value is set in <PropertySource*x*>*columnvalue*</PropertySource*x*> field.	String	Supports the use of all filename macros that can be used in t transport properties except for the **%Filename%** a **%Extension%** macros.

4. Use an expression shape in an orchestration to set the **Microsoft.XLANGs.BaseTypes.Address** property for the dynamic send port. This property is used to specify the URI that the dynamic send port routes the message to. For an example of how to set the **Microsoft.XLANGs.BaseTypes.Address** property for a dynamic send port see the **Create an expression** section of the topic Walkthrough: Module 3 - Accessing SharePoint Properties from an Orchestration ..

It is also possible to dynamically set certain properties of a late bound Windows Sharepoint Services send port in an orchestration. If this is done, the Windows SharePoint Services port will be configured twice, once through the Windows SharePoint Services context properties and once through the Windows SharePoint Services Transport Properties dialog box. By default, the configuration specified in the Windows SharePoint Services Transport Properties dialog box takes precedence over the configuration properties specified in the context properties. In order to use the configuration specified in the context properties follow these steps:

1. Follow the steps in the topic How to Create a Send Port to create a Static One-way send port.

2. When setting the properties for the send port, define the URI for the send port by entering the appropriate values for the **Sharepoint Site URL** and **Destination Folder URL** properties.

3. Set the value of the **Overwrite** property to **Orchestration** if you want to use the value defined by the context property **WSS.ConfigOverwrite** in an orchestration.

4. Set the **Microsoft Office Integration** property to **Orchestration** if you want to use the value defined by the context property **WSS.ConfigOfficeIntegration** in an orchestration.

5. Enter a value of **-1** for any send port properties that use the integer data type if you want to set those values with a context property in an orchestration.

6. Leave blank any send port properties that use the string data type if you want to set those values with a context property in an orchestration. This does not apply to the

Sharepoint Site URL and **Destination Folder URL** properties. These properties must be specified in the **Windows Sharepoint Services Transport Properties** dialog box.

7. Use a **Message Assignment** shape within a **Construct Message** shape in an orchestration to set the configuration properties for the outbound message. The **Construct a new message** section of this topic illustrates how to set configuration properties of an outbound message.

8. Any send port properties that are configured with a value of -1 (for properties that use the integer data type), "Orchestration" (for drop-down enumeration properties) or are left blank (for properties that use the string data type) will be set at run time with the context property that was specified in the orchestration.

To preserve InfoPath processing instructions for InfoPath forms with embedded attachments processed by BizTalk Server

If you use the Windows Sharepoint Services adapter to receive InfoPath forms with embedded attachments, complete the following steps to preserve any InfoPath processing instructions that are in the form:

1. If you are using a map in the orchestration to map data from one InfoPath form to another InfoPath form, ensure that you have set the **Copy Processing Instructions (PIs)** property in the map to **Yes**. This parameter is set under the **Custom Header** section of the **Grid Properties** page for the map.

2. If you are not using a map in the orchestration, update the output message using the following expression in a message assignment shape:

3. In the expression above, *NewMessage* is the output message that you are adding the processing instructions to.

Windows SharePoint Services Adapter Properties Reference

The following Windows SharePoint Services adapter properties are promoted into BizTalk Server or are used to specify send port configuration options for outgoing messages. The properties can be used to access Windows SharePoint Services information regarding the message or to provide information to the Windows SharePoint Services adapter from within an orchestration.

Message property precedence

There is a rule of precedence for overriding the message properties defined in orchestrations and send ports.

The following are the rules:

1. Property defined in the orchestration inside of PropertiesXML

2. Property defined in the orchestration

3. Property defined at the send port level inside of the Property Name/ or Property Source collection

4. Property defined at the send port level

Considerations and Known Issues

The following are considerations for the Windows SharePoint Services adapter properties:

- The list of properties in orchestrations is merged with the properties defined by the port based on property position. If there are conflicts, the orchestration property will override the send port property.

Property types

Property Type	Description
IN	IN properties are BizTalk Server properties that get their value from Windows SharePoint Services.
CONFIG	CONFIG properties are properties that get their value from BizTalk orchestrations or custom pipelines. This value is used by the Windows SharePoint Services adapter when determining the destination of the outgoing messages. CONFIG properties allow you to specify the value of some of the properties within an orchestration or custom pipeline that you would otherwise have to define on the send port. Properties that don't begin with IN or CONFIG are both IN and CONFIG, except for the URL property.
PROMOTED	PROMOTED properties can be used by content-based routing (CBR). Properties that are not marked as PROMOTED cannot be used by CBR.
SPECIAL	N/A

Property list

Windows SharePoint Services Standard Column	Windows SharePoint Services Property Name and Type	Type	Description	Property Type
Name	Filename	xs:string	The file name with the extension of the Windows SharePoint Services file. File names, including extensions, are unique within a document	IN/CONFIG/ PROMOTED

			library.	
N/A	Url	xs:string	The URL of the file.	IN/PROMOTED
N/A	TransmittedFileLocation	N/A	This property is used by Business Activity Monitoring (BAM) for integration purposes and is not available in orchestrations.	SPECIAL
N/A	InArchivedMsgUrl	xs:string	The URL of the file in the archive document library. This property is not available if the receive location is not archiving the message.	IN/PROMOTED
Type	InIconUrl	xs:string	The URL of the Windows SharePoint Services icon that is used to represent the document.	IN
Title	InTitle	xs:string	The title of the Windows SharePoint Service file. This is different from the file name. Titles don't have to be unique within a document library.	IN/PROMOTED
Modified	InLastModified	xs:dateTime	The last modified date of the Windows SharePoint Service.	IN/PROMOTED
Modified By	InLastModifiedBy	xs:string	The name of the last user that modified the file.	IN/PROMOTED
ID	InItemId	xs:int	The ID of the file. This is an integer unique within the document library which can be used to access the file.	IN

Edit	InEditUrl	xs:string	The URL that can be accessed to edit the properties of the file.	IN
Created	InCreated	xs:dateTime	The date when the Windows SharePoint Service file was created.	IN/PROMOTED
Created By	InCreatedBy	xs:string	The user that created the file.	IN/PROMOTED
File Size	InFileSize	xs:int	The size of the Windows SharePoint Services file.	IN
N/A	InListName	xs:string	The name of the document library where this file is located.	IN/PROMOTED
N/A	InListUrl	xs:string	The URL of the document library, or document library folder where this file is located.	IN
N/A	InPropertiesXml	xs:string	A flat XML document that contains all the standard and user defined Windows SharePoint Services columns. It allows access to any Windows SharePoint Services column value from an orchestration, including the values of the user-defined columns.	IN
N/A	InOfficeIntegration	xs:string	Based on the value of the receive location. This is either *yes*, *no*, or *optional*.	IN
N/A	ConfigOverwrite	xs:string	"Yes" overwrites the already existing files with the same name.	CONFIG

			"No" raises an error when a file with the same name exists. "Rename" changes the file to a unique name by appending a unique sequence to the file name.	
N/A	ConfigNamespaceAliases	xs:string	The alias definitions of the XPATHs.	CONFIG
N/A	ConfigOfficeIntegration	xs:string	'Yes' if the OfficeImporters should be called. 'No' to handle the message as-is. 'Optional' results in 'Yes' if IP solution is found, otherwise 'No'.	CONFIG
N/A	ConfigTemplatesDocLib	xs:string	Fallback document library name. This is the second place that is searched.	CONFIG
N/A	ConfigTemplatesNamespaceCol	xs:string	Namespace column name for fallback document library.	CONFIG
N/A	ConfigCustomTemplatesDocLib	xs:string	Primary document library name. This is the first place searched. **Note** This is similar to the Templates Document Library field for physical send ports.	CONFIG
N/A	ConfigCustomTemplatesNamespaceCol	xs:string	Namespace column name for primary document library.	CONFIG

N/A	ConfigPropertiesXml	xs:string	A flat XML document that contains all the Windows SharePoint Services column names and values that follow to be updated in Windows SharePoint Services. It allows an orchestration developer to set the SharePoint column values for the subsequent message to be created in SharePoint.	CONFIG
N/A	ConfigTimeout	xs:int	Time-out in milliseconds for Web service calls.	CONFIG
N/A	ConfigAdapterWSPort	xs:int	The port or IIS Web site where the adapter has been installed and configured.	CONFIG

Windows SharePoint Services Adapter Expressions

This topic describes the format and the meaning of the strings that can be specified as values for the **File Name Property Source** properties of the Windows SharePoint Services adapter. It also describes the related context properties, **WSS.Filename** and **WSS.ConfigPropertiesXml**. These expressions allow you to easily define the file name value, or custom Windows SharePoint Service column value, based on literals as well as values extracted from the message or the BizTalk system.

The expressions can contain literals and macros. The literals will show up in the file name exactly as you type them. Macros must be placed between '%' characters. An example of a macro is *%MessageID%* which at runtime will be replaced with the GUID of the message.

Expression examples

Design time value	Runtime value
XYZ	XYZ
PurchaseOrder	PurchaseOrder
%MessageID%	55B93F27-7455-4066-

	ABE1-B4EBE6839A1A
PurchaseOrder - %MessageID%	PurchaseOrder - 55B93F27-7455-4066-ABE1-B4EBE6839A1A
Discount \%10	Discount %10
PurchaseOrder - %XPATH=//ns0:PurchaseOrder/ns0:ID%	PurchaseOrder – 10001
PurchaseOrder - %XPATH=//ns0:PurchaseOrder/ns0:PartnerName%-%XPATH=//ns0:PurchaseOrder/ns0:ID%	PurchaseOrder – Contoso-10001

Supported macros

Design time value	Runtime value
%MessageID%	The BizTalk message ID which is a unique GUID.
%SendingOrchestrationID%	The BizTalk ID of the orchestration instance where the message originated.
%SendingOrchestrationType%	The type name of the orchestration where the message originated.
%XPATH=<xpath>%	Allows specifying an XPATH to be used for extracting the value from the message. “<xpath>” must be replaced with a valid XPATH expression.
%Filename%	Replaced with the filename value extracted from the message context property WSS.Filename. Messages received from SharePoint have the WSS.Filename context property value set to the name of the SharePoint file. The returned value is preprocessed using Path.GetFilenameWithoutExtension.
%Extension%	Replaced with the file extension value extracted from the message context property WSS.Filename. Messages received from SharePoint have the WSS.Filename context property value set to the name of the SharePoint file. The returned value is preprocessed using Path.GetExtension. The returned value will not contain ".".

Any valid expression supported by property promotion is a valid design time file name. The design time file name will be expanded at runtime into Windows SharePoint Services file

names. This Windows SharePoint Services file name has some additional limitations, which are described as follows:

- Valid Windows file names can contain any Unicode characters with the exception of the following: / \ : * ? < > | " # { } % & ~ or tab characters and multiple periods.
- The file name cannot be longer than 255 characters and the entire URL must be shorter than 255 characters.
- If the expanded Windows SharePoint Services file name contains invalid characters, or if the expanded file name or URL is too long, an error will be logged in the application event log and the message will be suspended. The error and the message state will also be visible in Health and Activity Tracking (HAT).

Supported Windows SharePoint Services Column Types

This topic describes the Windows SharePoint Services column types that are supported by the Windows SharePoint Services adapter. The values of these column types can be set in the message.

Supported types

The following table describes the supported column types:

UI Type	SharePoint Object Model Type	Sample	Comments
Single Line of Text	SPFieldType.Text	single line	
Multiple lines of text	SPFieldType.Note	line 1 line 2 line 3	
Choice Drop-Down	SPFieldType.Choice	ChoiceA	ChoiceA from the available choices (ChoiceA, ChoiceB, ChoiceC)
Choice Radio-Buttons	SPFieldType.Choice	#ChoiceB;#ChoiceC;#	ChoiceB and ChoiceC are enabled, ChoiceA is disabled (available choices are ChoiceA, ChoiceB, ChoiceC). Use ;# as a separator.
Number	SPFieldType.Number	123.456	

Currency USD (or any other)	SPFieldType.Currency	100.00	
Lookup	SPFieldType.Lookup	1	The number is the item identifier inside the referenced list.
YesNo	SPFieldType.Boolean	1	1=Yes 0=No
Hyperlink or Picture	SPFieldType.URL	http://www.microsoft.com, Microsoft Web Site	URL separated with "," from the display text. The "Microsoft Web Site" text will be a hyperlink to http://www.microsoft.com
Date only	SPFieldType.DateTime	2005-02-11T10:05:04	The DateTime as defined by the XML standard for the xs:dateTime. The pattern for dateTime is CCYY-MM-DDThh:mm:ss where CC represents the century, YY the year, MM the month, and DD the day, preceded by an optional leading negative (-) character to indicate a negative number. If the negative character is omitted, positive (+) is assumed. The T is the date/time separator and hh, mm, and ss represent hour, minute, and second, respectively. This representation may be immediately followed by a "Z" to indicate Coordinated Universal Time (UTC, also referred to as Greenwich Mean Time) or to indicate the time zone.
Date and Time/SPFieldType.DateTime	2005-02-11T10:05:04		The DateTime as defined by the XML standard for the xs:dateTime. The pattern for dateTime is CCYY-MM-DDThh:mm:ss

			where CC represents the century, YY the year, MM the month, and DD the day, preceded by an optional leading negative (-) character to indicate a negative number. If the negative character is omitted, positive (+) is assumed. The T is the date/time separator and hh, mm, and ss represent hour, minute, and second, respectively. This representation may be immediately followed by a "Z" to indicate UTC or to indicate the time zone.

Windows SharePoint Services Adapter Walkthroughs

This section consists of three walkthroughs. The first shows you how to configure BizTalk Server 2006 to send and receive messages using the Windows SharePoint Services adapter and content-based routing (CBR). The second shows you how to integrate BizTalk Server with Microsoft Office. The third shows you how to access the Windows SharePoint Services context properties of an incoming message at run time and then determine the destination of that message based on a property using dynamic ports in an orchestration.

In This Section

- Walkthrough: Module 1 - Sending and Receiving Messages with the Windows SharePoint Services Adapter
- Walkthrough: Module 2 - Integrating Office with the Windows SharePoint Services Adapter
- Walkthrough: Module 3 - Accessing SharePoint Properties from an Orchestration

Walkthrough: Module 1 - Sending and Receiving Messages with the Windows SharePoint Services Adapter

This walkthrough shows you how to configure Windows SharePoint Services and BizTalk Server so you can send and receive a message using the Windows SharePoint Services Adapter and content-based routing (CBR). Content-based routing eliminates the need for message subscription for messages that are deterministically bound to specific ports. It also provides additional flexibility for users who want to route messages based on envelope properties or simply based on receive port configuration properties. For an introduction to the Windows SharePoint Services adapter.

Prerequisites

The following are prerequisites for performing the procedures in this topic:

- You must have a single-server deployment with a complete installation of BizTalk Server 2006 running on Windows Server 2003.

For information about using the Windows SharePoint Services adapter in a multiserver deployment.

Configure Windows SharePoint Services

In this procedure you create a SharePoint top-level Web site that contains three document libraries. The Windows SharePoint Services adapter uses these libraries to move a message from a source library to a destination library. This message is also archived in a document library. This procedure must be done to provide the Windows Sharepoint Services site that is accessed by the Windows Sharepoint Services adapter in this walkthrough and to set user rights to enable access to this site.

Create a Windows SharePoint Services site

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **SharePoint Central Administration.**

2. Under **Virtual Server Configuration**, click **Create a top-level Web site**.

3. Under **Virtual Server List**, select the Web site that you installed the Windows SharePoint Services Adapter on. For example, *Default Web Site*.

4. In the **Web Site Address** section, in the **URL name** field, type *WSSAdapterWalkthrough*.

5. In the **Site Collection Owner** section, in the **User name field,** type a user name. This user will be the owner for the Web site and does not need special permissions in BizTalk Server.

6. In the **Site Collection Owner** section, in the **E-mail** field, type in an e-mail address.

7. Click **OK**.

8. On the **Top-Level Site Successfully Created** page, click the new top-level Web site you just created. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

9. Select the **Team Site** template from the list of templates, and then click **OK**. This will open the Team Web Site Home page.

Create a "Source" document library

1. On the Team Web Site Home page, on the top navigation bar, click **Create**.

2. Under **Document Libraries**, click **Document Library**.

3. In the **Name and Description** section, in the **Name field,** type *Source*.

4. In the **Navigation** section, select **Yes** to display this form library on the Quick Launch bar.

5. In the **Document Template** section, in the **Document Template** drop-down list, select *None*.

6. Click **Create**. The document library will be created and you will be redirected to the empty library.

Create a "Destination" document library

1. On the Team Web Site Home page, on the top navigation bar, click **Create**.

2. Under **Document Libraries**, click **Document Library**.

3. In the **Name and Description** section, in the **Name field**, type *Destination*.

4. In the **Navigation** section, select **Yes** to display this form library on the Quick Launch bar.

5. In the **Document Template** section, in the **Document Template** drop-down list, select *None*.

6. Click **Create**. The document library will be created and you will be redirected to the empty library.

Create an "Archive" document library

1. On the Team Web Site Home page, on the top navigation bar, click **Create**.

2. Under **Document Libraries**, click **Document Library**.

3. In the **Name and Description** section, in the **Name field**, type *Archive*.

4. In the **Navigation** section, select **Yes** to display this form library on the Quick Launch bar.

5. In the **Document Template** section, in the **Document Template** drop-down list, select *None*.

6. Click **Create**. The document library will be created and you will be redirected to the empty library.

7. Close the *WSSAdapterWalkthrough* Web site.

8. Close the **SharePoint Central Administration** Web site.

Configure Windows security

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Computer Management**.

2. In the console tree, expand **Local Users and Groups**, and then click **Groups**.

3. Right-click the **SharePoint Enabled Hosts** group, click **Add to Group**, and then click **Add**.

4. In the **Select Users, Computers, or Groups dialog box**, under **Enter the object names to select**, type the name of the account that you configured the BizTalk Server Host Instance to run under, and then click **OK**.

5. In the console tree, expand **Services and Applications**, and then click **Services**.

6. Right-click **BizTalk Service BizTalk Group: <BizTalk_Host_Name>**, and then click **Restart**.

7. Close **Computer Management**.

Configure SharePoint security

1. Open a Web browser and navigate to the URL of the site you created. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

2. On the Team Web Site Home page, on the top navigation bar, click **Site Settings**.

3. Under **Administration**, click **Manage users**.

4. Click **Add Users**.

5. In **Step 1: Choose Users**, type the name of the account that the BizTalk Server Host Instance is running under.

6. In **Step 2: Choose Site Groups**, select the **Reader** and **Contributor** check boxes.

7. Click **Next**.

8. Clear the **Send the following e-mail to let these users now they've been added** check box, and then click **Finish**.

9. Close the *WSSAdapterWalkthrough* Web site.

Create and configure the BizTalk Server ports

In this procedure you will create and configure the BizTalk Server receive ports, receive locations, and send ports for the Windows SharePoint Services adapter. These ports are points of entry into and out of BizTalk Server for documents received and sent by the Windows Sharepoint Services adapter.

Create the receive port

1. Click **Start**, **Programs**, **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration.**

2. Expand **Microsoft BizTalk Server 2006 Administration SnapIn**, expand **BizTalk Group**, expand **Applications**, expand **BizTalk Application 1**, right-click **Receive Ports**, click **New**, and then click **One-way Receive Port…**

3. In the **Receive Port Properties** dialog box, under **General**, type *FromSource* in the **Name** field.

4. Click **OK**.

Create the receive location

1. In the **BizTalk Administration Console**, right-click the **Receive Locations** node, click **New**, and then click **One-way Receive Location**.

2. In the **Select a Receive Port** dialog box, select *FromSource*, and then click **OK**.

3. In the **Receive Location Properties** dialog box, under **General**, type *SourceLocation* in the **Name** field.

4. In the **Transport** section, in the **Type** drop-down list, select *Windows SharePoint Services*.

5. Click **Configure** to configure the Windows SharePoint Services adapter properties.

6. In the **Adapter Web Service Port** property, type the port number of the virtual server where the Windows SharePoint Services adapter Web service was installed. By default, this is port 80.

7. Type *Archive* in the **Archive Location** property.

8. Type *10* in the **Polling Interval** property.

9. Type the URL to your SharePoint site in the **SharePoint Site Url** property. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

10. Type *Source* for the **Source Document Library** property.

11. Click **OK**.

12. In the **Receive Location Properties** dialog box, select *BizTalkServerApplication* as the **Receive handler**.

13. In the **Receive pipeline** drop-down list, select *PassThruReceive*.

14. Click **OK**.

Create the send port

1. In the **BizTalk Administration Console**, right-click the **Send Ports** node, click **New**, and then click **Static One-way Send Port**.

2. In the **Send Port Properties** dialog box, under **General**, type *SendToDestination* in the **Name** field.

3. In the **Transport** section, select *Windows SharePoint Services* for the type.

4. Click **Configure** to configure the Windows SharePoint Services adapter properties.

5. In the **Adapter Web Service Port** property, type the port number of the virtual server where the Windows SharePoint Services adapter Web service was installed. By default, this is port 80.

6. Type in *Destination* for the **Destination Folder** property.

7. Type in *PurchaseOrder1-%MessageID%.xml* for the **Filename** property.

8. Set the **Overwrite** property to *Yes*.

9. Type in the URL to your SharePoint site in the **SharePoint Site Url** property. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

10. Set the **Microsoft Office Integration** property to *No*.

11. Click **OK**.

12. In the **Send Port Properties dialog box**, in the **Send handler** drop-down list, select *BizTalkServerApplication*.

13. In the **Send pipeline** drop-down list, select *PassThruTransmit*.

14. Click the **Filters** tab.

15. Select *WSS.InListName* in the **Property** field.

16. Select == in the **Operator** field.

17. Type *Source* in the **Value** field.

18. Click **OK**.

Enable and start the receive location and receive port

In these procedures you enable the receive location and start the receive port. This procedure must be completed to allow the Windows Sharepoint Services adapter to send and receive messages through the specified send port and receive location.

Enable the receive location

1. In the **BizTalk Administration Console**, click the **Receive Locations** node.
2. Right-click *SourceLocation*, and then click **Enable**.

Start the send port

1. In the **BizTalk Administration Console**, click the **Send Ports** node.
2. Right-click *SendToDestination*, and then click **Start**.
3. Close the **BizTalk Administration Console**.

Sending a message through the system

In this procedure you create an XML document and upload it to the Windows SharePoint Services Web site. The Windows SharePoint Services adapter will take that message, archive it in the Archive document library, and then send it to the Destination document library. This procedure demonstrates how a document flows from a Sharepoint web site, through BizTalk Server, and to a Sharepoint Services Web site using the Windows Sharepoint Services adapter.

Create a working directory

1. Create a directory on your computer called **WSSAdapterWalkthrough**. For example, *C:\WSSAdapterWalkthrough*.

Create an XML file

1. Click **Start**, point to **Programs**, point to **Accessories**, and then click **Notepad.**
2. Type the following:
3. Save the file in your working directory as *PurchaseOrder1.xml*. For example, *C:\WSSAdapterWalkthrough\PurchaseOrder1.xml*.

Upload the XML file

1. Open a Web browser and navigate to the URL of the site you created in the last task. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.
2. On the left side, under **Documents**, click **Source**.
3. Click **Upload Document**.
4. In the **Name** box, type or browse to the XML file you created above. For example, *C:\WSSAdapterWalkthrough\PurchaseOrder1.xml*, and then click **Save and Close**. You should now be able to see the file in the list.

5. Refresh the browser window. The *PurchaseOrder1.xml* file will no longer be listed in this library.

6. On the top navigation bar, click **Documents and Lists**.

7. Under **Document Libraries**, click **Destination**.

8. In the Destination Document Library, you will now see your message listed. You will also find a copy archived in the Archive Document Library.

Walkthrough: Module 2 - Integrating Office with the Windows SharePoint Services Adapter

This walkthrough is a continuation of Walkthrough: Module 1 - Sending and Receiving Messages with the Windows SharePoint Services Adapter and shows you how to integrate Microsoft Office with the BizTalk Server content-based routing (CBR) application you created. For an introduction to the Windows SharePoint Services adapter see What Is the Windows SharePoint Services Adapter? .

Prerequisites

The following are prerequisites for performing the procedures in this topic:

- You must have a single-server deployment with a complete installation of BizTalk Server 2006 running on Windows Server 2003.

- You must complete the following walkthrough: Walkthrough: Module 1 - Sending and Receiving Messages with the Windows SharePoint Services Adapter

Create a BizTalk project

In this procedure you create an empty BizTalk project and a schema using the BizTalk Editor. This procedure is required to create the schema for the InfoPath form that is used later.

Create a strong name key file

1. Click **Start**, point to **Programs**, point to **Visual Studio 2005**, point to **Visual Studio Tools**, and then click **Visual Studio Command Prompt**.

2. Type *sn -k C:\WSSAdapterWalkthrough\OrderProcess.snk*, and then press **Enter**. The key pair will be written.

3. Close the command prompt.

Create an empty BizTalk project

1. Click **Start**, **Programs**, **Microsoft Visual Studio 2005**, and then click **Microsoft Visual Studio 2005**.

2. Click **File**, **New**, and then click **Project**.

3. Under **Project types**, select **BizTalk Projects**.

4. Under **Templates**, select **Empty BizTalk Server Project**.

5. Type *OrderProcess* in the **Name** field.

6. Type the file path to your working directory in the **Location** field. For example, *C:\WSSAdapterWalkthrough*.

7. Click **OK**.

Associate the key file with the assembly

1. In Solution Explorer, right-click the *OrderProcess* project, and then click **Properties**.

2. Under **Common Properties**, select **Assembly**.

3. Under **Strong Name**, in the **Assembly Key File** field, type *C:\WSSAdapterWalkthrough\OrderProcess.snk*.

4. Click **OK**.

Create an XSD schema by using the BizTalk Editor

1. In Solution Explorer, right-click the *OrderProcess* project, click **Add**, and then click **New Item**.

2. Under **Categories**, click **Schema Files**.

3. Under **Templates**, click **Schema**.

4. Type *OrderProcessSchema* in the **Name** field, and then click **Add**.

5. In the Properties Window for *OrderProcessSchema*, select *Qualified* for the **Element FormDefault** property.

6. In the Properties Window for OrderProcessSchema, type http://OrderProcess.PurchaseOrder in the **Target Namespace** field.

7. In the **BizTalk Editor**, right-click *Root*, click **Rename**, and then type *PurchaseOrder*.

8. Right-click the **PurchaseOrder** node, click **Insert Schema Node**, then click **Child Field Element**.

9. Name it *PurchaseOrderID*.

10. Create another child field element and name it *BillTo*.

11. Create another child field element and name it *Amount*.

12. In the Properties Window, set the **Data Type** property for *Amount* to xs:unsignedInt.

13. Create another child field element and name it *PurchaseOrderDate*.

14. In the Properties Window, set the **Data Type** property for *PurchaseOrderDate* to xs:dateTime.

15. Click **File**, and then click **Save All**.

16. Close **Microsoft Visual Studio 2005**.

Create an InfoPath form

In this procedure you create another document library and an InfoPath form based on the schema you created in the last procedure. This InfoPath form will be used to submit a document to BizTalk Server.

Create a new document library

1. Open a Web browser and navigate to the URL of the site you created. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

2. On the top navigation bar, click **Create**.

3. Under **Document Libraries**, click **Document Library**.

4. In the **Name and Description** section, type *InfoPathSolutions* in the **Name field**.

5. In the **Navigation** section, select **Yes** to display this form library on the Quick Launch bar.

6. In the **Document Template** section, select *None* for the **Document Template**.

7. Click **Create**. You will be redirected to the empty library you just created.

8. On the left side, click **Modify Settings and Columns**.

9. Under **Columns**, click **Add a New Column**.

10. Under **Name and Type**, type *Namespace* in the **Name** field.

11. Click **OK**.

12. Close the *WSSAdapterWalkthrough* Web site.

Create an InfoPath form based on the OrderProcessSchema schema file

1. Click **Start**, point to **Programs**, point to **Microsoft Office**, and then click **Microsoft Office InfoPath 2003**.

2. In the **Fill Out a Form** dialog box, select **Design a Form.**

3. In the **Design a Form** task pane, select **New from XML Document or Schema**.

4. In the **Data Source Wizard**, click **Browse** and select the schema file you created in the last procedure. For example, C:\WSSAdapterWalkthrough\OrderProcess\OrderProcess\OrderProcessSchema.xsd.

5. Click **Next**, and then click **Finish**.

6. In the **Data Source** task pane, right-click the **PurchaseOrder** node, and then click **Section with Controls**. This will create the form on the template.

7. Click **File**, click **Save**, and then click **Save**.

8. In the **Save As** dialog box, type *PurchaseOrder.xsn* in the **File name** field, and then click **Save**.

9. Click **File**, and then click **Publish**.

10. In the **Publishing Wizard**, click **Next**.

11. Select **To a Web Server**, and then click **Next**.

12. Type the path and filename to your InfoPathSolutions document library, and then click **Next**. For example, http://<server_name>/sites/WSSAdapterWalkthrough/InfoPathSolutions/PurchaseOrder.xsn.

13. Click **Finish**, and then click **Close**.

14. Close Microsoft InfoPath 2003.

Modify the SharePoint document libraries

In this procedure you will update the namespace property for the PurchaseOrder.xsn file and modify the Destination Document Library. This namespace is used as a variable when determining subscribers of published documents for content based routing scenarios.

Update the namespace for PurchaseOrder.xsn

1. Open a Web browser and navigate to the URL of the site you created. For example, http://<server_name>/sites/WSSAdapterWalkthrough.

2. On the left side, under **Documents**, click *InfoPathSolutions*.

3. Move the pointer over *PurchaseOrder.xsn*, right-click it, and then click **Edit Properties**.

4. Type http://OrderProcess.PurchaseOrder in the **Namespace** field, and then click **Save and Close**.

Modify the Destination document library

1. On the top navigation bar, click **Documents and Lists**.

2. Under **Document Libraries**, click **Destination**.

3. On the left side, click **Modify Settings and Columns**.

4. Under **Columns**, click **Add New Column**.

5. Under **Name and Type**, type *Partner Name* in the **Column name** field.

6. Click **OK**.

7. Close the *WSSAdapterWalkthrough* Web site.

Modify the send port from walkthrough 1

In this procedure you modify the send port from walkthrough 1. This procedure is required to ensure that the document processed in this walkthrough is correctly routed to the send port.

Modify the send port

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration.**

2. Expand **Microsoft BizTalk Server 2006 Administration SnapIn**, expand **BizTalk Group**, expand **Applications**, expand **BizTalk Application 1**, and then click the **Send Ports** node.

3. Right-click *SendToDestination*, and then click **Properties**.

4. Under **Transport**, click **Configure**.

5. In the **Filename** field, type PurchaseOrder2-%XPATH=//pons:PurchaseOrder/pons:PurchaseOrderID%.xml.

6. In the **Namespace Aliases** field, type pons="http://OrderProcess.PurchaseOrder".

7. In the **Templates Document Library**, type InfoPathSolutions.

8. In the **Templates Namespace Column**, type Namespace.

9. Select Yes for the **Microsoft Office Integration** property.

10. Under **Windows SharePoint Services Integration**, type Partner Name in the **Column 01** field.

11. Type %XPATH=//pons:PurchaseOrder/pons:BillTo% in the **Column 01 Value** field, click **OK**, and then click **OK** again to exit the **Send Port Properties** dialog box.

Restart the send port

1. In the **BizTalk Administration Console**, click the **Send Ports** node.

2. Right-click *SendToDestination*, and then click **Unenlist**.

3. Right-click *SendToDestination*, and then click **Start**.

4. Close the **BizTalk Administration Console**.

Send a message through the system

In this procedure you create an InfoPath form and upload it to the Windows SharePoint Services Web site. The Windows SharePoint Services adapter will take that message, archive it in the Archive document library, and then send it to the Destination document library. This procedure demonstrates how a document flows from a Sharepoint web site, through BizTalk Server, and to a Sharepoint Services Web site using the Windows Sharepoint Services adapter.

Create an InfoPath form to send through the system

1. Open a Web browser and navigate to the URL of the site you created. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

2. On the left side, under **Documents**, click *InfoPathSolutions*.

3. Click the *PurchaseOrder* file to display the **File Download** dialog box, and then click **Open**. InfoPath will load the form.

4. In the **Purchase Order ID** field, type *1002*.

5. In the **Bill To** field, type *John Doe*.

6. In the **Amount** field, type *750*.

7. In the **Purchase Order Date** field, type *1/2/2005*.

8. Click **Save**.

9. In the **Save As** dialog box, type *http://<server_name>/sites/WSSAdapterWalkthrough/Source* in the **file name** field, and then hit Enter.

10. Type *PurchaseOrder2.xml* in the **file name** field, and then click **Save**.

11. Close Microsoft Office InfoPath 2003.

12. In the Web browser, on the top navigation bar, click **Documents and Lists**.

13. Under **Document Libraries**, click **Destination**.

14. In the Destination document library, you will now see your message listed. You will also find a copy archived in the Archive document library.

15. In the Destination document library, click *PurchaseOrder1.xml*. Note that this XML file is opened in Microsoft Internet Explorer.

16. In the Destination document library, click *PurchaseOrder2.xml*. Note that this XML file is opened in Microsoft Office InfoPath 2003.

Walkthrough: Module 3 - Accessing SharePoint Properties from an Orchestration

This walkthrough is a continuation of Walkthrough: Module 2 - Integrating Office with the Windows SharePoint Services Adapter and shows you how to access the Windows SharePoint Services context properties of an incoming message at run time and then determine the destination of that message based on a property using dynamic ports in an orchestration. For an introduction to the Windows SharePoint Services adapter see What Is the Windows SharePoint Services Adapter? .

Prerequisites

The following are prerequisites for performing the procedures in this topic:

- You must have a single server deployment with a complete installation of BizTalk Server 2006 running on Windows Server 2003.

- You must complete the following walkthroughs: Walkthrough: Module 1 - Sending and Receiving Messages with the Windows SharePoint Services Adapter and Walkthrough: Module 2 - Integrating Office with the Windows SharePoint Services Adapter

For information about using the Windows SharePoint Services Adapter in a multi server deployment, see Setting Up and Deploying the Windows SharePoint Services Adapter.

Modify the BizTalk project

In this procedure you modify the PurchaseOrder schema from Walkthrough: Module 2 - Integrating Office with the Windows SharePoint Services Adapter . This procedure illustrates how to promote a schema property for easy access in a BizTalk orchestration.

Modify the PurchaseOrder.xsd schema

1. Click **Start**, point to **Programs**, point to **Microsoft Visual Studio 2005**, and then click **Microsoft Visual Studio 2005**.

2. Click **File**, click **Open**, and then click **Project/Solution**.

3. Browse to the *OrderProcess.sln* file, and then click **Open**.

4. In **Solution Explorer**, right-click the *OrderProcessSchema.xsd* file, and then click **Open**.

5. In **BizTalk Editor**, expand *PurchaseOrder*.

6. Right-click *Amount*, click **Promote**, and then click **Quick Promotion**.

7. Click **OK**.

8. Save *PurchaseOrder.xsd*.

Create an orchestration

In this procedure you create a new BizTalk orchestration. This procedure creates the orchestration that is used to process a message received by the Windows Sharepoint Services adapter.

Add a BizTalk orchestration

1. In **Solution Explorer**, right-click the *OrderProcess* project, click **Add**, and then click **New Item**.
2. Under **Categories**, select **Orchestration Files**.
3. Under **Templates**, select **BizTalk Orchestration**.
4. Type *MyCompanyOrderProcessing* in the **Name** field, and then click **Add**.

Create receive information

In this procedure you create a new message, receive port, and receive shape for the orchestration. This procedure illustrates how to configure an orchestration to receive a message from BizTalk Server.

Create a new message

1. In **Orchestration View**, right-click **Messages**, and then click **New Message**. This will generate a new message with the name *Message_1*.
2. Right-click *Message_1*, click **Rename**, and then type *Message_PO*.
3. Right-click *Message_PO*, and then click **Properties Window**.
4. In the **Message Type** property, expand **Schemas**, and then select *OrderProcess.OrderProcessSchema* schema.

Add a receive port to the orchestration

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Port** shape to the Port Surface. The Port Configuration Wizard will start.
2. On the Welcome screen, click **Next**.
3. Type *ReceivePurchaseOrder* in the **Name** field, and then click **Next**.
4. Select **Create a new Port Type**.
5. Type *PurchaseOrderPT* in the **Port Type Name** field, and then click **Next**.
6. On the **Port Binding screen**, leave the default values, and then click **Next**.

7. Click **Finish**.

8. In **Orchestration View**, under **Port Types**, expand the *PurchaseOrderPT* port type.

9. Right-click *Operation_1*, click **Rename**, and then type *PurchaseOrderOperation*.

Add a Receive shape to the orchestration

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Receive** shape to the Orchestration.

2. Right-click the Receive shape, and then click **Properties Window**.

3. Set the **Activate** property to *True*.

4. Type *Receive_PO* in the **Name** field.

5. In the **Properties Window**, select *Message_PO* for the Message property.

6. Select *ReceivePurchaseOrder.PurchaseOrderOperation.Request* for the **Operation** property. This will tie the port to the Receive shape in the Orchestration Designer.

Create send information

In this procedure you create a new message, send ports, and decision structure to the orchestration. This procedure illustrates how to configure an orchestration with decision logic and how to configure an orchestration to send a message to a send port.

Create a new message

1. In **Orchestration View**, right-click **Messages**, and then click **New Message**. This will generate a new message with the name *Message_1*.

2. Right-click *Message_1*, click **Rename**, and then type *Message_Task*.

3. Right-click *Message_Task*, and then click **Properties Window**.

4. In the **Message Type** property, expand **Schemas**, and then select *OrderProcess.OrderProcessSchema* schema.

Add a send port to the orchestration

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Port** shape to the Port Surface. The Port Configuration Wizard will start.

2. On the Welcome screen, click **Next**.

3. Type *SendPurchaseOrder* in the **Name** field, and then click **Next**.

4. Select **Use an existing Port Type**.

5. Under **Available Port Types**, select *OrderProcess.PurchaseOrderPT*, and then click **Next**.

6. On the **Port Binding screen**, under **Port direction of communication**, select *I'll always be sending messages on this port*, and then click **Next**.

7. Click **Finish**.

Add a Send shape to the orchestration

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Send** shape to the Orchestration Designer. Place it below the *Receive_PO* Receive shape.

2. Right-click the Send shape, and then click **Properties Window**.

3. Type *Send_PO* in the **Name** field.

4. Select *Message_PO* for the **Message** property.

5. Select *SendPurchaseOrder.PurchaseOrderOperation.Request* for the **Operation** property. This will tie the port to the Send shape in the Orchestration Designer.

Add a Decide shape to the orchestration

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Decide** shape to the Orchestration Designer. Place it below the *Send_PO* Send shape.

2. Right-click the Decide shape, and then click **Properties Window.**

3. Type *NeedsApproval* in the **Name** field.

4. In Orchestration Designer, click **Rule_1** on the Decide shape.

5. In the Properties Windows, type *ApprovalRequired* for the **Name** property.

6. Click the **Expression** property field, and then click the ellipsis (**...**) button.

7. In the BizTalk Expression Editor, type or copy the following:

8. Click **OK**.

Add another send port to the orchestration

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Port** shape to the Port Surface. The Port Configuration Wizard will start.

2. On the Welcome screen, click **Next**.

3. Type *SendToTasksList* in the **Name** field, and then click **Next**.

4. Select **Use an existing Port Type**.

5. Under **Available Port Types**, select *OrderProcess.PurchaseOrderPT*, and then click **Next**.

6. On the **Port Binding screen**, under **Port direction of communication**, select *I'll always be sending messages on this port*.

7. Under **Port binding**, select *Dynamic*, and then click **Next**.

8. Click **Finish**.

Add a Send shape to the Decide shape

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Send** shape to the Orchestration Designer. Place it below the *ApprovalRequired* shape.

2. Right-click the Send shape, and then click **Properties Window**

3. Type *CreateApprovalTask* in the **Name** field.

4. Select *Message_Task* for the **Message** property.

5. Select *SendToTasksList.PurchaseOrderOperation.Request* for the **Operation** property. This will tie the port to the Send shape in the Orchestration Designer.

Create an expression

In this procedure you add an Expression shape to your solution which assigns the Tasks path value to a variable. This procedure illustrates how to add logic to an orchestration to modify the properties of a dynamic send port.

Create a new expression

1. Under **BizTalk Orchestrations** in the Toolbox, drag an **Expression** shape before the *CreateApprovalTask* Send shape.

2. Right-click the Expression shape, and then click **Properties Window.**

3. Type *SetPortDestination* in the **Name** field.

4. Click the **Expression** property field, and then click the ellipsis (**...**) button.

5. In the **BizTalk Expression Editor**, type the following:

6. Click **OK**.

Construct a new message

In this procedure you add a Construct shape to the solution which will construct a new instance of a message type within the orchestration. This procedure illustrates how to create a new message that is a copy of the inbound message and then modify the context properties

of the new message. This step is required because messages are immutable in BizTalk; that is, once you have constructed it, you cannot modify the original.

Add a Construct Shape

1. Under **BizTalk Orchestrations** in the Toolbox, drag a **Construct Message** shape before the *SetPortDestination* Expression shape.

2. Right-click the Construct Message shape, and then click **Properties Window.**

3. Type *ConstructTaskMessage* in the **Name** field.

4. Select *Message_Task* for the **Messages Constructed** property.

5. Under **BizTalk Orchestrations** in the Toolbox, drag a **Message Assignment** shape into the *ConstructTaskMessage* **Construct Message** shape.

6. In the **Properties Window**, type *InitTaskMessage* in the **Name** field.

7. Click the **Expression** property field, and then click the ellipsis (**...**) button.

8. In the **BizTalk Expression Editor**, type or copy the following:

9. Click **OK**.

10. Click **File**, and then click **Save All**.

Build the BizTalk project

In this procedure you build and deploy the BizTalk project. This step is required to create and deploy the assembly that BizTalk Server uses at runtime.

Build and deploy the solution

1. Click **Build**, and then click **Build OrderProcess**.

2. Click **Build**, and then click **Deploy OrderProcess**.

3. Close Microsoft Visual Studio 2005.

Modify the receive location and send port

In this procedure you modify the existing receive location and send port to use XML processing for the pipelines. The receive XML pipeline persists message properties used during orchestration processing and the send XML pipeline persists the message properties that were applied in the orchestration which are subsequently used for message routing.

Modify the receive location

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration.**

2. Expand **Microsoft BizTalk Server 2006 Administration SnapIn**, expand **BizTalk Group**, expand **Applications**, expand **BizTalk Application 1**, and then click the **Receive Locations** node.

3. Right-click *SourceLocation*, and then click **Properties**.

4. In the **Receive Location Properties** dialog box, under **General**, select *XMLReceive* for the **Receive pipeline** property.

5. Click **OK**.

Modify the send port

1. Click the **Send Ports** node.

2. Right-click *SendToDestination*, and then click **Properties**.

3. In the **Send Port Properties** dialog box, under **General**, select *XMLTransmit* for the **Send pipeline** property.

4. Select the **Filters** tab.

5. Select the existing condition, press DELETE, and then click **OK**.

Start a new send port

1. Click the **Send Ports** node.

2. Right-click *OrderProcess_1.0.0.0_OrderProcess.MyCompanyOrderProcess_SendToTasksList_<GUID>*, and then click **Start**.

Bind the orchestration

In this procedure you bind the orchestration to the specified ports. This procedure is required to tie physical ports to the orchestration that you built and deployed.

Bind the orchestration

1. In the **BizTalk Administration Console**, click the **Orchestrations** node.

2. Right-click the *OrderProcess.MyCompanyOrderProcessing* orchestration, and then click **Properties**.

3. Select the **Bindings** tab.

4. Under **Host**, select *BizTalkServerApplication* in the **Host** field.

5. Under **Bindings**, select *FromSource* for the *ReceivePurchaseOrder* Inbound Logical Port.

6. Under **Bindings**, select *SendToDestination* for the *SendPurchaseOrder* Outbound Logical Port.

7. Click **OK**.

8. Right click *OrderProcess.MyCompanyOrderProcessing* orchestration, and then click **Start**.

Send a message through the system

In this procedure you create an InfoPath form and upload it to the Windows SharePoint Services Web site. The Windows SharePoint Services adapter will take that message, archive it in the Archive document library, and then send it to the Destination document library. During the processing of this message, Windows SharePoint Services context properties will be accessed that help determine the destination.

Create an InfoPath form to send through the system

1. Open a Web browser and navigate to the URL of the site you created. For example, *http://<server_name>/sites/WSSAdapterWalkthrough*.

2. In the Quick Launch menu, click *InfoPathSolutions*.

3. Click the *PurchaseOrder* file to display the **File Download** dialog box, and then click **Open**. InfoPath will load the form.

4. In the **Purchase Order ID** field, type *1003*.

5. In the **Bill To** field, type *John Doe*.

6. In the **Amount** field, type *1750*.

7. In the **Purchase Order Date** field, type *1/3/2005*.

8. Click **Save**.

9. In the **Save As** dialog box, type *http://<server_name>/sites/WSSAdapterWalkthrough/Source* in the **file name** field, and then press ENTER.

10. Type *PurchaseOrder3.xml* in the **file name** field, and then click **Save**.

11. Close InfoPath.

12. In the Web browser, click **Documents and Lists**.

13. Under **Document Libraries**, click **Destination**.

14. In the Destination document library, you will now see your message listed in this library. You will also find a copy archived in the Archive document library.

15. Click **Home**.

16. Under **Lists**, click **Tasks**.

17. In the Tasks list you will see the newly created approval task.

Creating and Deleting Adapter Handlers

This section describes how to create and delete adapter handlers in the BizTalk Server Administration Console. The procedures provided in this section are generic to multiple native adapters. This section does not contain procedures specific to a single adapter. For more information, see **Creating an FTP Receive Handler Using WMI**.

In This Section

- What Is an Adapter Handler?
- How to Create an Adapter Handler
- How to Delete an Adapter Handler

What Is an Adapter Handler?

An adapter handler is responsible for executing the adapter and contains properties for a specific instance of an adapter. A default BizTalk Server configuration will create adapter handlers for all of the installed adapters, but you may want to create additional adapter handlers for purposes of load balancing or to provide process isolation for a particular adapter handler.

Adapter handlers are bound to a BizTalk Host instance, and BizTalk Host instances are bound to a BizTalk server. Therefore, you must add additional BizTalk servers to your BizTalk group if you want to load balance adapter processing across BizTalk servers. You do not need to add additional BizTalk servers to your BizTalk group if you are creating additional adapter handlers for the purpose of process isolation.

If you need to create a new host instance to run an adapter handler in, you must first create a host and then create an instance of that host to run on one of your BizTalk servers..

All adapter handlers except for HTTP and SOAP adapter receive handlers must be configured to run in an in-process host. HTTP and SOAP adapter receive handlers can only be run in an isolated host.

How to Create an Adapter Handler

You can create a send or receive adapter handler by using the BizTalk Server Administration Console.

To create an adapter handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, right-click the adapter for which you would like to add a send or receive handler, point to **New**, and then click **Send Handler** to create a send handler or click **Receive Handler** to create a receive handler.

3. In the **<host name> Properties** dialog box, on the **General** tab, in the **Host Name** list, select the host with which the adapter handler will be associated.

4. If you are creating an adapter send handler, select the option to **Make this the default handler** if you would like for this to be the default send handler for this adapter.

5. Click **OK**.

How to Delete an Adapter Handler

You can delete a send or receive adapter handler by using the BizTalk Server Administration Console.

Before you remove an adapter handler, you must remove all receive locations or send ports with which it is associated.

To delete an adapter handler

1. In the BizTalk Server Administration Console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Adapters**.

2. In the expanded adapter list, select the adapter for which you want to delete the adapter handler.

3. Right-click the adapter handler that you want to delete, and then click **Delete**.

4. Click **Yes** to confirm that you want to delete this adapter handler.

5. Click **OK**.

Using the BizTalk Adapter Trace Utility

This topic describes how to install the Microsoft BizTalk Adapter Trace Utility. It also discusses how to run the BizTalk Adapter Trace Utility and how to have the generated trace file analyzed.

Install the Trace Utility

To install the BizTalk Adapter Trace Utility, follow these steps:

1. To download the Tracelog.exe file, visit the following Microsoft Platform SDK download Web site:

   http://go.microsoft.com/fwlink/?LinkId=21975

2. Start the Platform SDK Web installation program by clicking the link for the **PSDK-x86.exe** file at the bottom of the Web page.

3. When you are prompted, choose the option for a custom installation.

4. In the **Custom Installation** dialog box, click to clear all the available features.

5. Expand the **Microsoft Windows Core SDK** feature, and then expand the **Tools** feature.

6. Choose the **Tools (Intel 64-bit)** feature, and then click **Will be installed on local hard drive.**

7. Click **Next**, and then click **Next** again to start the installation.

8. Locate the *Drive:\MicrosoftPlatformSDKInstallationFolder*\bin folder, and then copy the Tracelog.exe file to the Microsoft BizTalk Server 2006 installation folder. The BizTalk Server 2006 installation folder also contains the Trace.cmd file.

Enable the BizTalk Adapter Trace Utility

To enable the BizTalk Adapter Trace Utility in BizTalk Server 2006, follow these steps:

1. At a command prompt, change the current directory to the directory where BizTalk Server 2006 is installed. By default, BizTalk Server 2006 is installed in the Program Files\Microsoft BizTalk Server 2006 directory.

2. Type the following command, and then press ENTER:

   trace -tools "Path of the BizTalk Adapter Trace Utility"

   By default, the BizTalk Adapter Trace Utility is located in the C:\Program Files\Microsoft Platform SDK\Bin directory. You must enclose the path of the BizTalk Adapter Trace Utility in quotation marks.

   For example, type the following command:

   trace -tools "C:\Program Files\Microsoft Platform SDK\Bin"

   The **-tools** switch indicates to the Trace.cmd file the location of the Tracelog.exe file.

Run the BizTalk Adapter Trace Utility

To run the BizTalk Adapter Trace Utility on a scenario, follow these steps:

1. At a command prompt, type the following command, and then press ENTER:

   trace -start

2. Reproduce the scenario that you want to trace.

3. At a command prompt, type the following command, and then press ENTER:

   trace -stop

4. After you stop the trace, a binary file that is named Bts2006.bin is generated in the folder where BizTalk Server 2006 is installed.

5. You can send the Bts2006.bin file to Microsoft Product Support Services for analysis.

How to Start, Stop, Pause, Resume, or Restart BizTalk Server Services

The following table lists the BizTalk Server services that you can start, stop, pause, resume, or restart:

Name	Description	Startup Type	Dependencies
BizTalk Base EDI Service	Processes EDI documents.	Automatic	None
BizTalk Service BizTalk Group: BizTalkServerApplication	Provides the BizTalk Server application service.	Automatic	BizTalk Base EDI Service Enterprise Single Sign-On (SSO) Service Event Log Remote Procedure Call (RPC)
Enterprise Single Sign-On Service	Provides single sign-on services to enterprise applications.	Automatic	None
Rule Engine Update Service	Notifies users about the deployment or undeployment of policies.	Automatic	None

Prerequisites

To perform this procedure, you must be a member of the Administrators group on the local computer, or you must have been delegated the appropriate authority. If the computer is joined to a domain, members of the Domain Admins group might be able to perform this procedure. As a security best practice, consider using Run as to perform this procedure.

To start, stop, pause, resume, or restart a BizTalk Server service

You can start, stop, pause, resume, or restart a BizTalk Server service by using any of the methods listed below:

- Using Services in Control Panel
- Using a command prompt

Using Services in Control Panel

1. Open Services. Click **Start**, click **Run**, and then type **services.msc**.
2. Right-click the appropriate BizTalk Server service and then click **Start**, **Stop**, **Pause**, **Resume**, or **Restart**.

Using a command prompt

1. Open Command Prompt. Click **Start**, click **Run**, and then type **cmd**.
2. Type one of the following, where ServiceName is the name of the BizTalk Server service you want to start, stop, pause, or resume:
   - To start a service, type:

     net start ServiceName
   - To stop a service, type:

     net stop ServiceName
   - To pause a service, type:

     net pause ServiceName
   - To resume a service, type:

     net continue ServiceName

BizTalk Server Service	ServiceName
BizTalk Base EDI Service	"edi subsystem"
BizTalk Service BizTalk Group: BizTalkServerApplication	btssvc$biztalkserverapplication
Enterprise Single Sign-On Service	Entsso
Rule Engine Update Service	Ruleengineupdateservice