Microsoft BizTalk Server 2006 Part-VI

Table of Contents

Creating a BAM Activity Definition with ODBA

You can define a Business Activity Monitoring (BAM) activity in an orchestration diagram. A BAM activity contains business data items of interest and milestones that you want to monitor. For example, you may want to know if a purchase order arrives, or if a certain number of purchase orders take too much time to fulfill. For information about BAM activities, see What Are BAM Activities.

The following table shows the steps you must complete to define a BAM activity in an orchestration diagram.

Step	Topic
1. Define milestones.	How to Define Milestones with ODBA
2. Define business data items.	How to Define Data of Interest with ODBA
3. Name the activity.	How to Set BAM Web Page Preferences with ODBA
4. Export the activity definition.	How to Export a BAM Activity Definition from ODBA

In This Section

- How to Define Milestones with ODBA
- How to Define Data of Interest with ODBA
- How to Set BAM Web Page Preferences with ODBA
- How to Export a BAM Activity Definition from ODBA
- How to View the Live BAM Web Page with ODBA

How to Define Milestones with ODBA

You define milestones on connectors in a business process orchestration diagram. Milestones are the transitions between actions in the orchestration diagram. For example, in a purchase order process, a "received" milestone would occur between the Begin shape and the Evaluation shape.

To define a milestone with ODBA

1. In the orchestration diagram, right-click a connector, and then click **Add Milestone**.
2. In the **Milestone Properties** dialog box, do the following:

Use this	To do this
Name	Type a name for the milestone.
Description	Type a description for the milestone.

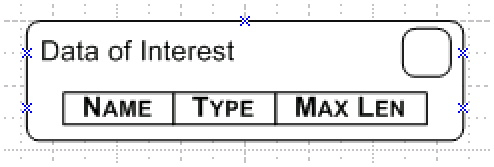
3. Click **OK**

How to Define Data of Interest with ODBA

Business data items together with milestones become the Business Activity Monitoring (BAM) activity definition, when you export the business process orchestration diagram as a BAM activity definition.

Business data items are data of interest that you want to track for the business process. You add business data items to a **Data of Interest** shape. You can add only one **Data of Interest** shape in a business process orchestration diagram. The **Data of Interest** shape is not part of the process flow of the diagram.

The following figure shows the **Data of Interest** shape.



For information about BAM activities and activity items, see What Are BAM Activities.

For information about defining milestones in ODBA, see How to Define Milestones with ODBA .

For information about exporting a BAM activity definition from a business process orchestration diagram, see How to Export a BAM Activity Definition from ODBA.

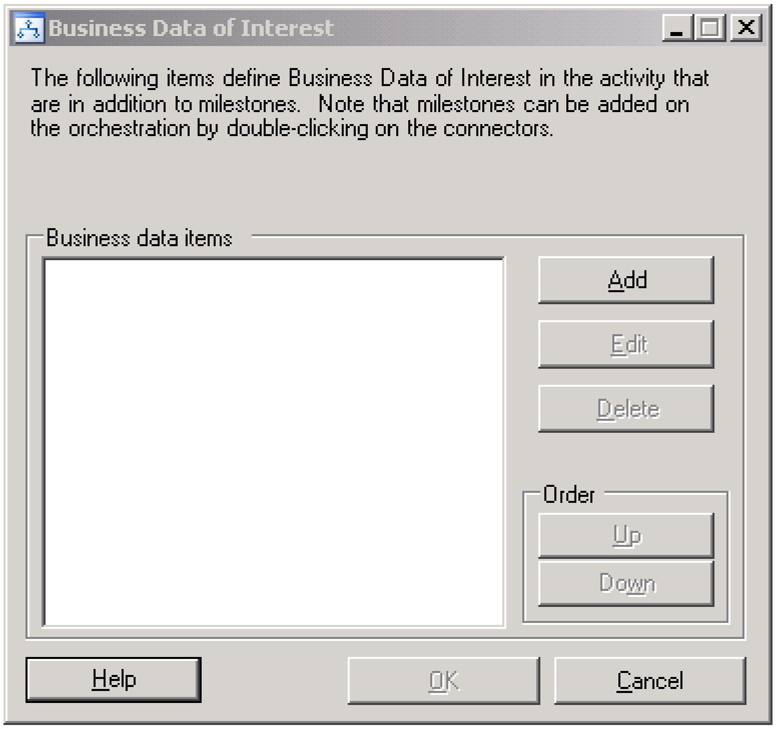
To define business data items

1. On the **ODBA** menu, click **Show/Hide Data of Interest Shape**.

2. Double-click the **Data of Interest** shape.

   Alternatively, you can right-click the **Data of Interest** shape, and then select **Edit Business Data Items**.

   The following figure shows the **Business Data of Interest** dialog box.

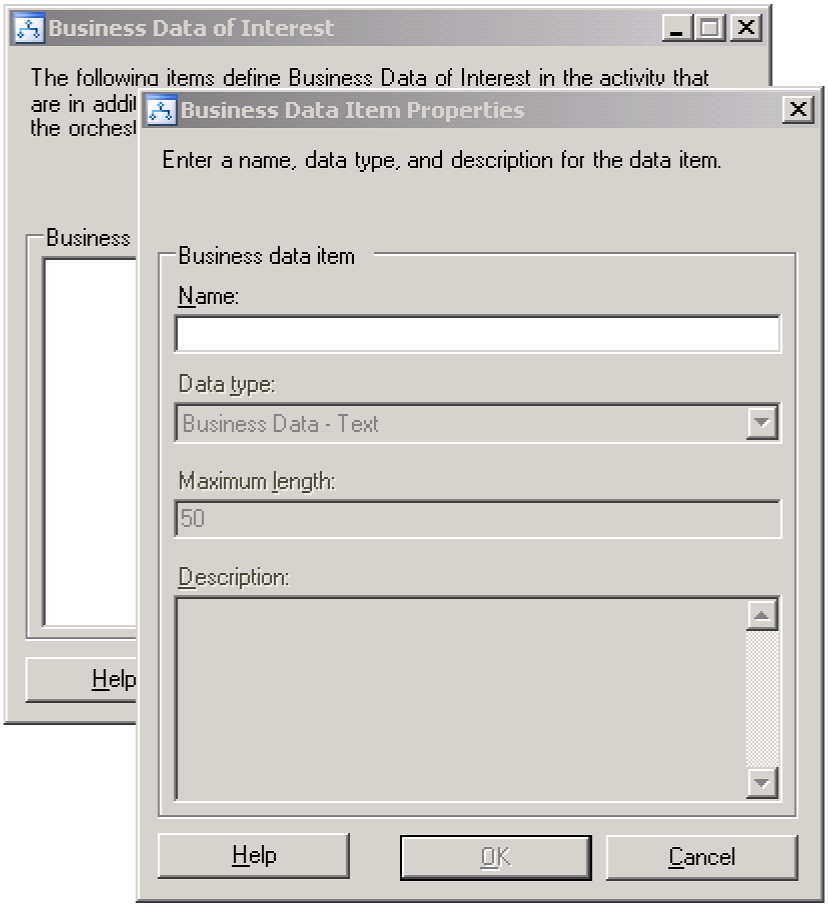


Business Data of Interest

The following items define Business Data of Interest in the activity that are in addition to milestones. Note that milestones can be added on the orchestration by double-clicking on the connectors.

3. In the **Business Data of Interest** dialog box, click **Add**.

The following figure shows the **Business Data Item Properties** dialog box.



The following items define Business Data of Interest in the activity that are in addi the orches

Enter a name, data type, and description for the data item.

Name:

Data type: Business Data - Text

Maximum length: 50

Description:

4. In the **Business Data Item Properties** dialog box, do the following:

Use this	To do this
Name	Type a name for the activity item.
Data type	From the drop-down list, select the data type for the item. Data types include: • Text • Integer • Decimal
Description	Type a description for the activity item.

5. Click **OK**.

6. Repeat steps 3 through 5 for each activity item you want to add.
7. In the **Business Data of Interest** dialog box, click **OK**.

Next Steps

After you define the business data items, you must name the activity. There are two ways to name a BAM activity in ODBA. You can name the activity when you set the BAM Web page preferences. Otherwise, BAM uses the file name of the .XML file to which you export the activity definition.

For information about naming the activity when you set the BAM Web page preferences, see How to Set BAM Web Page Preferences with ODBA.

For information about naming the activity when you export the BAM activity definition, see How to Export a BAM Activity Definition from ODBA.

How to Set BAM Web Page Preferences with ODBA

You can use Orchestration Designer for Business Analysts (ODBA) as a starting point to view Business Activity Monitoring (BAM) data. This makes the business process orchestration diagram useful throughout the business lifecycle. You use the business process orchestration diagram to understand the business process that is in progress. You access the BAM Web pages to see data about the instance of the process. When a process occurs multiple times, each time the process occurs is called an instance. You must configure your BAM Web page preferences before you can open a BAM Web page from ODBA.

To set BAM Web page preferences

1. On the ODBA menu, click **Set BAM Web Page Preferences**.
2. In the BAM Web Page Preferences dialog box, do the following:

Use this	To do this
Activity name	Type the name of the activity.
Web page URI	Type the address for the BAM Web page.
Preferred view	From the drop-down list, select the deployed view that you want to view in the BAM Web page.

3. Click **OK**.

After you set the BAM Web page preferences, you can access the live BAM Web page that you specified.

How to Export a BAM Activity Definition from ODBA

After you define Business Activity Monitoring (BAM) data of interest and milestones in a business process orchestration diagram, you export a BAM activity definition (.xml file) from the diagram. You can then import the activity definition into the BAM Add-In for Excel.

To export a BAM activity definition

1. On the **ODBA** menu, click **Export BAM Definition File**.

2. In the **Export BAM Definition** dialog box, select the first page of the business process orchestration diagram design that contains an activity definition you want to export, and then click **Browse**.

3. In the **Save BAM Definition File** dialog box, do the following:

Use this	To do this
Save in	Navigate to the folder where you want to save the BAM definition file.
File name	Type the file name you want to use for the BAM definition file. If you did not use the BAM Web Page Preferences to name the activity, BAM uses the export file name as the activity name.

4. Click **Save**.

After you export a BAM definition file, you can import it into a BAM workbook. For information about importing a BAM definition file into the BAM Add-In for Excel, see How to Import an Activity Definition from ODBA.

How to View the Live BAM Web Page with ODBA

After you set your Business Process Monitoring (BAM) Web page preferences, from ODBA, you can access the BAM portal to view live data.

Prerequisites

The following conditions must be met before you can access the BAM portal from ODBA:

- The BAM portal must exist. You can use ODBA without installing BizTalk Server. BAM is an optional feature of BizTalk Server, and BizTalk Server 2006 with BAM must be installed and configured so that the BAM portal exists.

- A BAM definition must be deployed. Deploying a BAM definition creates the pages and content in the BAM portal.

- You must set the BAM Web page preferences which include the location of the BAM portal.

- You must have permission to access the BAM portal.

To open the BAM Web page

1. On the **ODBA** menu, click **Go to Live BAM Web Page**.

Operations

BizTalk Server 2006 includes new features that make administering BizTalk Server quicker and easier than in previous releases of BizTalk Server. This section provides information about administering BizTalk Server.

In This Section

- Managing BizTalk Server

- Monitoring BizTalk Server

- Securing BizTalk Server

- Maintaining BizTalk Server

Managing BizTalk Server

This section provides information about how to manage BizTalk Server, and how to deploy and manage BizTalk applications.

In This Section

- Administration Tools

- BizTalk Server Administration Best Practices

- Using the BizTalk Server Administration Console

- Managing Groups

- Managing BizTalk Applications

- Managing Parties
- Managing Platform Settings
- How to Start, Stop, Pause, Resume, or Restart BizTalk Server Services

Administration Tools

You can use the following tools to administer BizTalk Server, to manage BizTalk Server groups, to deploy BizTalk Server applications, to interact with trading partners, and to monitor the status of BizTalk Server:

- **BizTalk Server Administration Console**. The BizTalk Server Administration Console is the primary management tool for BizTalk Server. It provides a graphical user interface for performing all of the deployment operations for a BizTalk application. For example, you can start the Import, Installation, and Export Wizards as well as add and remove an application's artifacts and make other modifications to the application. For information about using the BizTalk Administration console, see Using the BizTalk Server Administration Console.

- **BTSTask command-line tool**. New with BizTalk Server 2006, **BTSTask** enables you to perform many administrative tasks from the command line. For more information about using BTSTask, see BTSTask Command-Line Reference.

- **BTSdeploy command-line tool**. BTSdeploy enables you to deploy and remove assemblies from the BizTalk Management database. It also allows you to import and export bindings. BTSdeploy is included in BizTalk Server 2006 to support scripts that administrators have developed to use with previous versions of BizTalk Server. While you can continue to use your BTSdeploy scripts, you should use BTSTask commands when creating new scripts. You should also consider converting your existing scripts to use BTSTask, as BTSdeploy may be removed in subsequent versions of BizTalk Server. For more information about using BTSdeploy, see BTSDeploy Command-Line Reference (Deprecated).

- **Scripting and programmability APIs**. You can use Microsoft Windows Management Instrumentation (WMI) or the BizTalk Explorer Object Model to create and run scripts that automate administrative tasks. For information about using WMI, see **WMI Class Reference** . For more information about the BizTalk Explorer Object Model, see Using the BizTalk Explorer Object Model from Managed Code

  The WMI object model exposes and simplifies administrative APIs. All administration APIs expose some form of the following operations on every object they manage: create, enumerate, modify, and delete. WMI exposes this functionality in a consistent manner for all WMI objects.

- **BizTalk Explorer**. Solutions developers use BizTalk Explorer in Microsoft Visual Studio .NET to perform administrative tasks. BizTalk Explorer provides developers working in a single-computer environment access to administration tools. For information about using BizTalk Explorer in Visual Studio .NET, see Managing Business Processes Using BizTalk Explorer .

- **Business Activity Services (BAS) Web site.** The Business Activity Services (BAS) Web site is a Microsoft Windows SharePoint Services Web service site that provides business users with a simple-to-use interface that they can use to interact with trading partners and business processes in a familiar environment. For information about administering the BAS Web site, see Managing Business Activity Services .

- **Business Activity Monitoring (BAM).** BAM uses a Microsoft Office Excel workbook to provide business users with a way to see a real-time holistic view of business processes. Administrators manage the following:

  - **Business Activity Monitoring (BAM) dynamic infrastructure**. You can use the BAM Management utility to manage the BAM dynamic infrastructure. For information about managing the BAM dynamic infrastructure, see Managing the BAM Dynamic Infrastructure .

  - **BAM Event Bus Service**. You can use the BAM Event Bus Service snap-in to manage the BAM Event Bus Service. For information about managing the BAM Event Bus Service, see Managing the Tracking Data Decode Services (TDDS) .

  - **BAM configuration schema**. You can update the BAMConfiguration.xml file to set values for BAM database and Data Transformation Services parameters. For information about the BAM configuration schema, see BAM Configuration Schema .

  - **BAM Query Web service**. You can use the BAM Web service to query individual instance data stored in the BizTalk Tracking database. For information about updating the BAM Query Web service password, see **Managing the BAM Query Web Service**.

- **Health and Activity Tracking (HAT)**. You can use the Health and Activity Tracking (HAT) tool to track the health of your BizTalk Server implementation, identify bottlenecks, and monitor the BizTalk Server environment. You can view the technical details of a particular orchestration, pipeline, or message instance, as well as see the message flow of a particular message that enters the system.

  Business users can view, monitor, and query the tracked data. In addition, users can create custom views and queries, and save them for reuse in other locations. Business analysts and end users can track the state of their business processes by viewing both live and archived data. For more information about HAT, see Health and Activity Tracking.

- **Human Workflow Services**. You can use the Human Workflow Services (HWS) Administration console in BizTalk Server to manage human workflow objects. For information about using the HWS Administration console, see Using the

HWS Administration Console .

You can use WMI to create and run scripts that perform HWS administrative tasks. For information about using WMI to administration HWS, see **Human Workflow Services Classes** .

Best Practices for Deploying a BizTalk Application

This topic describes best practices for deploying a BizTalk application.

Group related artifacts together in a single application

As much as possible, place related artifacts in the same BizTalk application. This allows you to manage and deploy the artifacts as a single entity, making management easier. You can group artifacts that support the same business process or artifacts that perform similar functions into a single application.

Deploy shared artifacts in a separate application

If artifacts are going to be shared by two or more applications, deploy the shared artifacts into a separate application. For example, if two applications share a schema, place the schema in a separate application. This is because only one artifact having the same locally unique identifier (LUID) -- which consists of the artifact name and optionally other attributes -- can exist in a BizTalk group. If you include an artifact in one application, and then create a reference to it from another application, you may encounter problems such as the referring application not functioning correctly when you stop the application containing the artifact.

This best practice applies to all artifact types except for files, such as Readme files and scripts, which are added to the application as a File type of artifact. This is because more than one file artifact having the same name can be deployed in a BizTalk group. Therefore, you can use a file having the same name in two or more applications. Stopping one application will not impact the other application. For more information about adding File artifacts, see How to Add a File to an Application.

For best practices on sharing specific artifact types, see "Deploy a shared Web site in a separate application," "Deploy shared policies in a separate application," and "Deploy shared certificates in a separate application" in this section.⊟Deploy a shared Web site in a separate application

If a Web site will be shared by more than one business solution, deploy the Web site in a separate application. This is because when you uninstall a BizTalk application, the virtual directory of any Web site that is part of the application is removed, even if the Web site is running. If the Web site is shared with another business solution, the other business solution will no longer function correctly as a result.

Deploy shared policies in a separate application

If a policy is used by two or more applications, you should deploy it in a separate application rather than creating a reference from one application to another. This is because when you stop an application, its policies are undeployed. If you stop an application that includes a policy used by another application, the policy will no longer function in either application.

Deploy shared certificates in a separate application

If a certificate is used by a send port or receive location in two or more applications, you should deploy the certificate in a separate application, and then reference this application from the applications that need to use the certificate. This is because only one artifact having a particular LUID can exist in the BizTalk group, so you will not be able to import the same certificate in two different applications. If you attempt to import two applications that each use the same certificate, the first import will succeed, and the second will not. In this case, using the Overwrite import option does not correct the problem, as the existing certificate that you want to overwrite is contained in another application.

Never deploy an assembly from Visual Studio on a production computer

During the development process, the developer often must redeploy assemblies from Visual Studio. To enable the redeployment, Visual Studio may undeploy, unbind, stop, and unenlist artifacts included in the assemblies. Although this is necessary and appropriate in the development environment, it can cause unexpected and undesired consequences in a production environment. For this reason, as well as to avoid the possibility of anyone's attempting to deploy an assembly from Visual Studio on a production computer, we recommend that you never install Visual Studio on a production computer.

In addition, never refer to a production database from a computer running Visual Studio.

Using the BizTalk Server Administration Console

The BizTalk Server Administration Console is a Microsoft Management Console (MMC) that you can use to manage and monitor BizTalk Server, and that you can use to deploy and manage your BizTalk applications. The left side of the BizTalk Server Administration Console, the console tree, consists of folders and subfolders that represent different types of artifacts that you can manage.

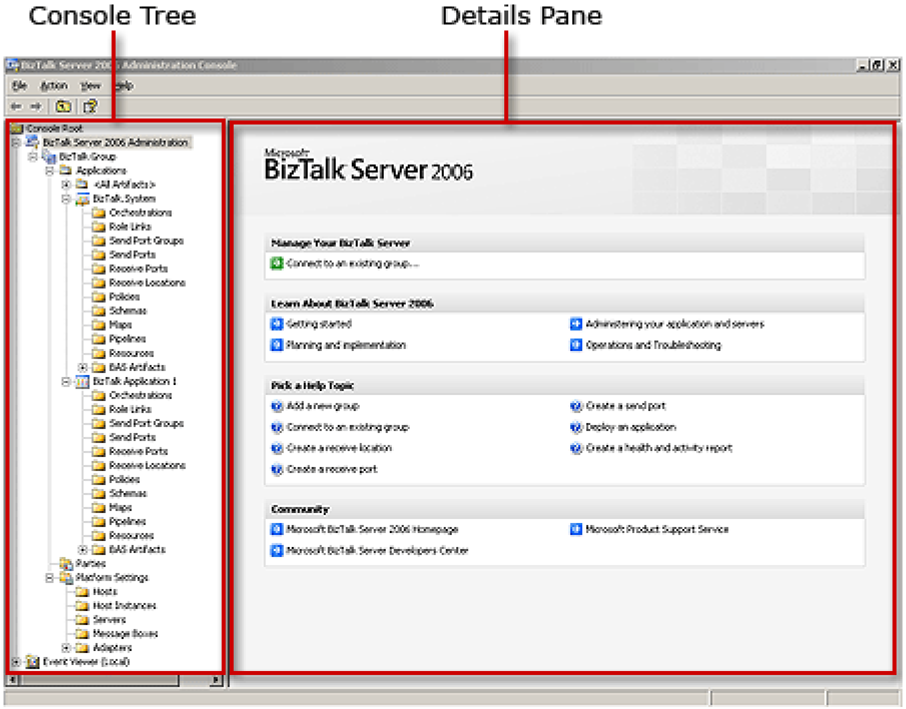
When you select a node in the console tree, the details pane on the right side of the Administration Console displays information about the items.

Selecting the BizTalk Server 2006 Administration node in the console tree displays the start page, which contains actions you can perform, such as connecting to an existing

BizTalk Server group. In addition the start page includes links to the BizTalk Server documentation and online community Web sites.

The following figure shows the BizTalk Server Administration Console with the startup hub page.

BizTalk Server Administration Console



BizTalk Server artifacts

You can manage the following artifacts using the BizTalk Server Administration Console:

- **BizTalk Group**. The BizTalk Group node in the console tree contains additional nodes that represent the artifacts (applications, parties, and platform settings) for that BizTalk group. BizTalk groups are units of organization that usually represent enterprises, departments, hubs, or other business units that require a contained BizTalk Server implementation. A BizTalk group has a one-to-one relationship with a BizTalk Management database. For more information, see Managing Groups.

- **Orchestration**. An orchestration is designed by using Orchestration Designer and is deployed to the BizTalk group under which it appears in the Administration console. For more information, see Managing Orchestrations.

- **Role Links**. A role link defines the relationship between roles by defined by the message and port types used in the interactions in both directions. For more information, see Managing Role Links.

- **Send port groups**. A send port group is a named collection of send ports that you can use to send the same message to multiple destinations in a single configuration. For more information, see Managing Send Ports and Send Port Groups.

- **Send ports**. A send port is a BizTalk object that sends messages. For more information, see Managing Send Ports and Send Port Groups.

- **Receive ports**. A receive port is a logical grouping of similar receive locations. For more information, see Managing Receive Ports.

- **Receive locations**. A receive location is defined as a specific address at which inbound documents arrive combined with a BizTalk Server pipeline that processes the messages received at that address. For more information, see Managing Receive Locations.

- **Policies**. A policy is a versioned collection of business rules. For more information, see Managing Policies.

- **Schemas**. A schema is the structure for a message. A schema can contain multiple subschema. For more information, see Managing Schemas.

- **Maps**. A map is an XML file that defines the correspondence between the records and fields in one specification and the records and fields in another specification. A map contains an Extensible Stylesheet Language (XSL) stylesheet that is used by BizTalk Server to perform the transformation described in the map. For more information, see Managing Maps.

- **Pipelines**. A pipeline is a software infrastructure that defines and links one or more processing stages, running them in prescribed order to complete a specific task. Pipelines divide processing into stages, abstractions that describe a category of work. They also determine the sequence in which each category of work is performed. For more information, see Managing Pipelines.

- **Resources**. A resource is a script, deployed assembly, or other file associated with an application.

- **BAS Artifacts**. Business Activity Services (BAS) artifacts provide a way for applications to communicate with outside partners. BAS artifacts are made up of partner profiles (also known as parties), partner groups, agreements, and Microsoft Office InfoPath templates. BAS artifacts must first be created using the BAS site before they can be imported and appear in the BizTalk Server Administration Console.

- **Parties**. A party is an entity outside of BizTalk Server that interacts with an orchestration. All of the partners your organization deals with are considered parties, and your organization may have several thousand partners. For more information, see Managing Parties.

- **Platform Settings**. The Platform Settings node contains subnodes for hosts, host instances, MessageBox databases, and adapters. For more information, see Managing Platform Settings.

  - **Hosts**. The Hosts node contains all of the in-process and isolated hosts in the BizTalk Server environment. A BizTalk Host is a logical container for items such as adapter handlers, receive locations (including pipelines), and orchestrations. For more information, see Managing BizTalk Hosts and Host Instances.

  - **Host Instances**. The Host Instances node contains all of the host instances in the current BizTalk Server group. Host instances are processes of the BizTalk Server runtime that execute your application components.

    Using the Host Instances node, you can create new host instances and refresh host instance information. For more information, see Managing BizTalk Hosts and Host Instances.

  - **Servers**. The Servers node lists all servers that are joined to the BizTalk Server group. These are the computers where BizTalk Server is installed and configured, and where host instances are running. Host instances are created by associating a server with a particular host.

  - **Message Boxes**. The Message Boxes node contains all MessageBox databases used by the current BizTalk Server group. Using the Message Boxes node, you can create new MessageBox databases and refresh MessageBox database information. The MessageBox database is the basis for work item load balancing across servers that do cooperative processing. A work item can pass through a MessageBox database more than once during its processing life. The name of the MessageBox database cannot exceed 100 characters. For more information, see Managing MessageBox Databases.

  - **Adapters**. The Adapters node contains subnodes for all of the send and receive adapters configured for the BizTalk Server group and the associated adapter handlers. Adapters are the messaging middleware used to send and receive messages between endpoints. For more information, see Using Adapters .

BizTalk Server Group Hub page

Selecting the BizTalk Group node displays the BizTalk Server Group Hub page in the details pane. The BizTalk Server Group Hub page is divided into three sections that provide an overall view of the health of your BizTalk Server system:

- The **Configuration Overview** section, located in the upper part of the Group Hub page, indicates the overall health of the BizTalk group by displaying the state of the applications, host instances, and adapter handlers configured in this group.

- The **Work in Progress and Suspended Items** sections are located in the middle of the Group Hub page.

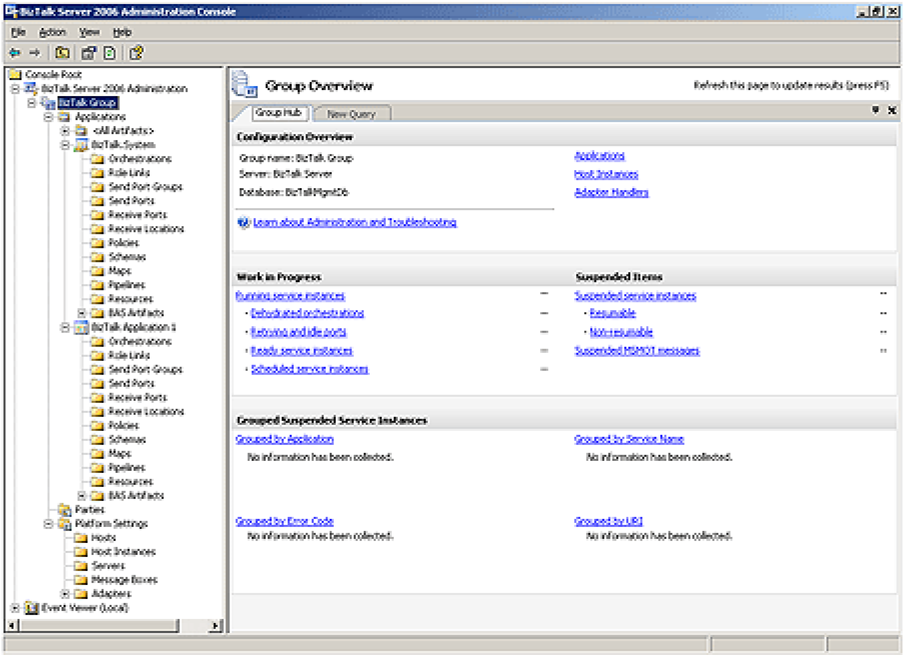
  - The **Work in Progress** section displays running service instances, dehydrated orchestrations, retrying and idle ports, ready service instances, and scheduled service instances.

  - The **Suspended Items** section displays the number of suspended service instances, resumable and non-resumable instances, and suspended MSMQT messages.

- The **Grouped Suspended Service Instances** section, located in the lower part of the Group Hub page, displays suspended service instances grouped by application, service name, error code, and URI.

The following figure shows the BizTalk Server Group Hub page.

BizTalk Server Group Hub page



Refresh this page to update results (press F5)

Configuration Overview

Group name: BizTalk Group
Server: BizTalk Server
Database: BizTalkMgmtDb

Applications
Host Instances
Adapter Handlers

Learn about Administration and Troubleshooting

Work in Progress

Running service instances	--
· Dehydrated orchestrations	--
· Retrying and idle ports	--
· Ready service instances	--
· Scheduled service instances	--

Suspended Items

Suspended service instances	--
· Resumable	--
· Non-resumable	--
Suspended MSMQT messages	--

Grouped Suspended Service Instances

Grouped by Application
No information has been collected.

Grouped by Service Name
No information has been collected.

Grouped by Error Code
No information has been collected.

Grouped by URI
No information has been collected.

For information about using the keyboard shortcuts for this feature, see Administration Console Keyboard Shortcuts.

How to Open the BizTalk Server Administration Console

You can use the following procedures to open the BizTalk Server Administration Console, and to sort or refresh the data displayed in the details pane.

Prerequisites

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

To open the BizTalk Server Administration Console

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

  Or

- Click **Start**, click **Run**, type **Btsmmc.msc**, and then click **OK**.

To refresh the BizTalk Server Administration Console

- Right-click the node in the console tree you want to refresh and click **Refresh**.

  This refreshes all the objects within that node.

To sort columns in the details pane of the BizTalk Server Administration Console

- In the console tree, click an item to view its information in the details pane.
- In the details pane, click the column heading by which you want to sort. This sorts the items under the heading in ascending order.
- Click the column heading again to sort in the descending (opposite) order.

To reorder the columns in the details pane of the BizTalk Server Administration Console

- In the details pane, click and drag the column to move the column to a new location to change the order in which the columns appear in the details pane.

Using the Administration Console Query Tab

You can use the Query tab in the BizTalk Server Administration Console to search for and locate specific running and suspended service instances, messages, or subscriptions. Queries performed using the Administration Console locate live items, which are stored in the MessageBox database. A new query tab appears each time you run a new query.

To locate archived messages or service instances, you use Health and Activity Tracking (HAT). For more information, see Health and Activity Tracking.

In This Section

- How to Save a Query
- How to Open a Saved Query
- How to Search for All Service Instances
- How to Search for Running Service Instances
- How to Search for Suspended Service Instances
- How to Search for Messages
- How to Search for Subscriptions

How to Save a Query

You can save a query for later reuse. By default, saved queries are stored in the **My Documents** folder of the user who created and saved them.

Prerequisites

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

To save a query

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**, and then click the BizTalk group.
3. In the details pane, click the **New Query** tab.
4. Create a query as desired, and then click **Save As**.
5. In the **Save As** dialog box, browse to or create the folder where you want to save the query.
6. In **File name**, type a name for the query, and then click **Save**.

How to Open a Saved Query

You can open a saved query directly from the folder where the query is saved or from within the BizTalk Server Administration Console. When you open a query directly from a folder, it opens the appropriate BizTalk Server group in the Administration Console. When you open a query from within the Administration Console, you have the option of running the query on any of the groups listed.

Prerequisites

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

To open a saved query from disk

1. Browse to the folder where the saved query is stored.

2. Double-click the saved query, which opens the BizTalk Server Administration Console and executes the query.

To open a saved query from the Administration Console

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**, and then click the BizTalk group.

3. In the details pane, click the **New Query** tab, and then click **Open Query**.

4. In the **Open** dialog box, browse to the saved query that you want to open, select that query, and then click **OK**.

How to Search for All Service Instances

You can use the **Query** tab in the BizTalk Server Administration Console to search for all service instances. You cannot unenlist a specific service type if there are any running or suspended instances. Before you can unenlist a specific service type, you can search for all service instances to ensure that there are no running or suspended instances.

Prerequisites

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

To search for all service instances

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**, and then click the BizTalk group.

3. In the details pane, click the **New Query** tab.

4. In the **Query Expression** group, in the **Value** column, select **All Service Instances** from the drop-down list box.

5. In the **Field Name** column, in the empty drop-down list box next to the asterisk (*****), select one or more of the following:

Item	**Description**
Application Name	The BizTalk Server application.
Creation Time	Find all service instances created before or after the specified date.
Group Results By	You can group results by application, host name, service class, service instance status, or service name.
Host Name	The name of the BizTalk Host.
Instance Status	You can search for all running instances, all suspended instances, active instances, dehydrated instances, ready to run instances, scheduled instances, suspended but not resumable instances, or suspended and resumable instances.
Maximum Matches	The number of matches to display.
Service Class	You can search for isolated adapters; messaging; messaging, MSMQT, and isolated adapters; MSMQT; Orchestration; or Routing Failure Report.
Service Instance ID	You can group or filter service instances by service instance ID.
Service Name	You can group or filter service instances by service name.
Service Type ID	You can group or filter service instances by service type ID.

6. Complete the **Value** column as appropriate for the selection you made in the **Field Name** column.

7. Continue adding additional lines to the query as appropriate, by completing the **Field Name**, **Operator**, and **Values** columns, and then click **Run Query**.

How to Search for Running Service Instances

You can use the **Query** tab in the BizTalk Server Administration Console to search for a specific dehydrated service instance.

Prerequisites

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

To search for running service instances

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**, and then click the BizTalk group.

3. In the details pane, click the **New Query** tab.

4. In the **Query Expression** group, in the **Value** column, select **Subscriptions** from the drop-down list box.

5. In the **Field Name** column, in the empty drop-down list box next to the asterisk (*****), select one or more of the following:

Item	**Description**
Maximum Matches	The number of matches to display.
Service Instance ID	You can group or filter running services instances by service instance ID.
Service Name	You can group or filter running service instances by service name.
Subscription Type	You can group or filter running service instances by Activation Subscription or Instance Subscription.

6. Complete the **Value** column as appropriate for the selection you made in the **Field Name** column.

7. Continue adding additional lines to the query as appropriate, by completing the **Field Name**, **Operator**, and **Values** columns, and then click **Run Query**.

How to Search for Suspended Service Instances

You can use the **Query** tab in the BizTalk Server Administration Console to search for suspended service instances. You can search for a specific sub-set of messages to locate a specific message associated with a service name, type, host, etc.

Prerequisites

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

To search for suspended service instances

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**, and then click the BizTalk group.

3. In the details pane, click the **New Query** tab.

4. In the **Query Expression** group, in the **Value** column, select **Suspended Service Instances** from the drop-down list box.

5. In the **Field Name** column, in the empty drop-down list box next to the asterisk (*****), select one or more of the following:

Item	**Description**
Application Name	The name of the BizTalk Server application.
Creation Time	Find suspended service instances created before or after the specified date.
Error Adapter	You can group or filter suspended service instances by adapter type: EDI, File, FTP, HTTP, MQSeries, MSMQ, POP3, SMTP, SOAP, SQL, or Windows SharePoint Services.
Error Code	You can group or filter suspended service instances by error code to show all that service instances have been suspended with that error code.
Error Description	You can group or filter suspended service instances with the specified error description.
Group Results	You can group or filter results by adapter, application, error code, error

By	description, host name, service class, service instance status, service name, or URI.
Host Name	Group or filter suspended service instances by host name.
Instance Status	You can search for suspended but not resumable instances, or suspended and resumable instances.
Maximum Matches	The number of matches to display.
Service Class	You can search for isolated adapters; messaging; messaging, MSMQT, and isolated adapters; MSMQT; Orchestration; or Routing Failure Report.
Service Name	You can group or filter suspended service instances by service name.
Service Type ID	You can group or filter suspended service instances by service type ID.
Suspension Time	You can group or filter suspended service instances suspended before or after the specified date.
URI	You can group or filter suspended service instances by URI.

6. Complete the **Value** column as appropriate for the selection you made in the **Field Name** column.

7. Continue adding additional lines to the query as appropriate, by completing the **Field Name**, **Operator**, and **Values** columns, and then click **Run Query**.

How to Search for Messages

You can use the **Query** tab in the BizTalk Server Administration Console to search for a specific class of messages. Searching for specific messages is especially useful when you are troubleshooting a suspended message on a receive port that uses the MSMQT adapter.

Prerequisites

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

To search for messages

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**, and then click the BizTalk group.

3. In the details pane, click the **New Query** tab.

4. In the **Query Expression** group, in the **Value** column, select **Messages** from the drop-down list box.

5. In the **Field Name** column, in the empty drop-down list box next to the asterisk (*****), select one or more of the following:

Item	**Description**
Creation Time	Find messages created before or after the specified date.
Error Adapter	You can group or filter messages by adapter type: EDI, file, FTP, HTTP, MQSeries, MSMQ, POP3, SMTP, SOAP, SQL, or Windows SharePoint Services.
Error Code	You can group or filter messages by error code.
Error Description	You can group or filter messages by error description.
Host Name	You can group or filter messages by host name.
Instance Status	You can search for all of the following types of instances: all running instances, all suspended instances, active instances, dehydrated instances, ready-to-run instances, scheduled instances, suspended but not resumable instances, or suspended and resumable instances.
Maximum Matches	The number of matches to display.
Message ID	You can group or filter messages by message ID.
Message Status	You can search for messages with consumed, in process, suspended, suspended but not resumable, suspended and resumable, undelivered, undelivered but scheduled, and undelivered but waiting for retry status.
Message Type	You can group or filter messages by message type.
Service Class	You can search for isolated adapters; messaging; messaging, MSMQT, and isolated adapters; MSMQT; Orchestration; or Routing Failure Report.
Service	You can group or filter messages by service instance ID.

Instance ID	
Service Name	You can group or filter messages by service name.
Service Type ID	You can group or filter messages by service type ID.
URI	You can group or filter messages by URI.

6. Complete the **Value** column as appropriate for the selection you made in the **Field Name** column.

7. Continue adding additional lines to the query as appropriate, by completing the **Field Name**, **Operator**, and **Values** columns, and then click **Run Query**.

How to Search for Subscriptions

You can use the **Query** tab in the BizTalk Server Administration Console to search for subscriptions. This is useful when you want to review all of the subscriptions defined in the system. When troubleshooting routing failures, you can review subscriptions to see if any of them are improperly configured and thus, causing the routing failure.

Prerequisites

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

To search for subscriptions

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**, and then click the BizTalk group.

3. In the details pane, click the **New Query** tab.

4. In the **Query Expression** group, in the **Value** column, select **Subscriptions** from the drop-down list box.

5. In the **Field Name** column, in the empty drop-down list box next to the asterisk (*****), select one or more of the following:

Item	**Description**
Maximum Matches	The number of matches to display.

Service Instance ID	You can group or filter subscriptions by service instance ID.
Service Name	You can group or filter subscriptions by service name.
Subscription Type	You can group or filter subscriptions by Activation Subscription or Instance Subscription.

6. Complete the **Value** column as appropriate for the selection you made in the **Field Name** column.

7. Continue adding additional lines to the query as appropriate, by completing the **Field Name**, **Operator**, and **Values** columns, and then click **Run Query**

Managing Groups

This section provides procedures for managing a BizTalk group. You use the BizTalk Server group to represent a unit of organization—such as an enterprise, department, or hub—that requires a contained Microsoft BizTalk Server 2006 implementation. The BizTalk Management (BizTalkMgmtDb) database stores configuration information for the BizTalk group and the servers in that group.

In This Section

- How to Connect to an Existing Group
- How to Reconnect to a Group
- 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 Modify Group Properties

How to Connect to an Existing Group

You can use the BizTalk Server 2006 Administration Console on any computer in your enterprise to remotely manage one or more BizTalk Server groups within your enterprise, as long as these groups are located on computers within the same domain or within trusted domains.

The BizTalk Server 2006 Administration node in the BizTalk Server Administration Console is the highest-level node for all BizTalk groups, and is the level that you use when adding an existing BizTalk Server group to the BizTalk Server Administration

Console. When you add a group, you must specify an existing server and BizTalk Management database to which you want to connect.

Prerequisites

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

To connect to an existing group

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

2. In the console tree, right-click **BizTalk Server 2006 Administration**, and then click **Connect to Existing Group**.

3. In the **Connect to Existing BizTalk Server Configuration Database** dialog box, in the **SQL Server name** drop-down list box, select the name of the Microsoft SQL Server instance that hosts the BizTalk Management database (also referred to as the Configuration database). When you select the instance of SQL Server, BizTalk Server automatically attempts to detect BizTalk Server databases on that computer.

4. In the **Database name** drop-down list box, select the BizTalk Server Management database (**BizTalkMgmtDb**) to which you want to connect, and then click **OK**.

   The BizTalk Server Administration Console adds the BizTalk Server group to the console tree.

How to Reconnect to a Group

group that you previously connected to your BizTalk Server may sometimes appear as unavailable in the BizTalk Server 2006 Administration Console. For example, a group can become unavailable to the Administration Console when the computer where the group resides is restarted. You can reconnect to an unavailable BizTalk Server group by using **Refresh** if the group is unavailable.

Prerequisites

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

To reconnect to a group

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

2. In the console tree, double-click **BizTalk Server 2006 Administration**.

3. Right-click the **(Unavailable)** group, and then click **Refresh**.

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,

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.

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

8. 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**.

9. 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

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

Before removing a server, you must remove all host instances on that server. For more information,

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**.

How to Modify Group Properties

You can use this procedure to configure global properties for your BizTalk Server group so that you can select a signing certificate, modify the cache refresh interval, and determine how BizTalk Server will handle large messages. For more information about BizTalk Server group properties.

Prerequisites

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

To configure BizTalk group 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, right-click **BizTalk Group**, and then click **Properties**.
3. In the **Group Properties** dialog box, on the **General** tab, do the following:

Use this	To do this
Name	Type the group name. The group name cannot exceed 128 characters in length. The name in this box appears next to the **Group** node in the console tree, along with the server name and the BizTalk Management database name.
Server	Displays the server on which the BizTalk Management database is physically stored.
Database	Displays the BizTalk Management database on which all configuration settings for this group are stored.
SSO Server name	Type the name of the Enterprise Single Sign-On (SSO) server that this computer will use to store and access adapter-specific information. This is the name of the SSO server used to connect to the SSO database.

BizTalk Administrator Group	Displays the name of the administrators group that has rights to manage the BizTalk Server environment after installation.
BizTalk Operators Group	Displays the name of the operators group that has rights to view configuration, run queries, and perform computer maintenance and basic application monitoring.
Cache Refresh (seconds)	Type or select the time (in seconds) that all items in the BizTalk Server administration cache must wait between configuration cache refreshes. Items involved in the refresh include the MessageBox databases, server properties, adapters, and connections to the Tracking database. By default, all objects in the cache are refreshed every 60 seconds, except for the server database connections and server properties.
Large message threshold (bytes)	Type or select the size at which BizTalk Server will fragment a message when the message size equals or exceeds this threshold size. The default threshold size is 1,000,000 bytes.
Large message fragment size (bytes)	Type or select the size of the fragments into which BizTalk Server will divide a large message when its size equals or exceeds threshold size. The default threshold size is 102,400 bytes.

4. On the **Databases** tab, do the following:

Use this	To do this
Databases	Displays the names of all databases in the application and the servers on which they reside, as specified during configuration.

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

Use this	To do this
Common Name	Displays a description of the selected certificate.
Thumbprint	Displays the thumbprint of the private key certificate that is used to digitally sign outbound messages from this group. 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 for the current user or personal store you want to use with the group.

Managing BizTalk Applications

This section provides information about managing BizTalk applications, including how to deploy, modify, update, and undeploy them. BizTalk applications are new with BizTalk Server 2006. They provide a way to view and manage the items, called "artifacts," that make up a BizTalk business solution. Examples of artifacts are BizTalk assemblies, .NET assemblies, schemas, maps, bindings, and certificates.

You can create a BizTalk application and add artifacts to it. You can then view, package, deploy, and manage the application and its artifacts as a single entity from within the BizTalk Administration console or by using the BTSTask command-line tool.

For background information about BizTalk application deployment and maintenance, see Understanding BizTalk Application Deployment and Management. For step-by-step instructions on deploying a simple application, which will familiarize you with the new application deployment features of BizTalk Server 2006, see Walkthrough: Deploying a Basic BizTalk Application.

In This Section

- Best Practices for Deploying a BizTalk Application
- BizTalk Application Deployment and Management Checklists
- BizTalk Application Deployment and Management Walkthroughs
- Understanding BizTalk Application Deployment and Management
- How to Start and Stop a BizTalk Application
- Creating and Modifying BizTalk Applications
- Deploying BizTalk Applications
- Managing Artifacts
- Undeploying BizTalk Applications

Best Practices for Deploying a BizTalk Application

This topic describes best practices for deploying a BizTalk application.

Group related artifacts together in a single application

As much as possible, place related artifacts in the same BizTalk application. This allows you to manage and deploy the artifacts as a single entity, making management easier. You can group artifacts that support the same business process or artifacts that perform similar functions into a single application.

Deploy shared artifacts in a separate application

If artifacts are going to be shared by two or more applications, deploy the shared artifacts into a separate application. For example, if two applications share a schema, place the schema in a separate application. This is because only one artifact having the same locally unique identifier (LUID) -- which consists of the artifact name and optionally other attributes -- can exist in a BizTalk group. If you include an artifact in one application, and then create a reference to it from another application, you may encounter problems such as the referring application not functioning correctly when you stop the application containing the artifact.

This best practice applies to all artifact types except for files, such as Readme files and scripts, which are added to the application as a File type of artifact. This is because more than one file artifact having the same name can be deployed in a BizTalk group. Therefore, you can use a file having the same name in two or more applications. Stopping one application will not impact the other application. For more information about adding File artifacts, see How to Add a File to an Application.

For best practices on sharing specific artifact types, see "Deploy a shared Web site in a separate application," "Deploy shared policies in a separate application," and "Deploy shared certificates in a separate application" in this section.

Deploy a shared Web site in a separate application

If a Web site will be shared by more than one business solution, deploy the Web site in a separate application. This is because when you uninstall a BizTalk application, the virtual directory of any Web site that is part of the application is removed, even if the Web site is running. If the Web site is shared with another business solution, the other business solution will no longer function correctly as a result.

Deploy shared policies in a separate application

If a policy is used by two or more applications, you should deploy it in a separate application rather than creating a reference from one application to another. This is because when you stop an application, its policies are undeployed. If you stop an application that includes a policy used by another application, the policy will no longer function in either application.

Deploy shared certificates in a separate application

If a certificate is used by a send port or receive location in two or more applications, you should deploy the certificate in a separate application, and then reference this application from the applications that need to use the certificate. This is because only one artifact having a particular LUID can exist in the BizTalk group, so you will not be able to import the same certificate in two different applications. If you attempt to import two applications that each use the same certificate, the first import will succeed, and the second will not. In this case, using the Overwrite import option does not correct the problem, as the existing certificate that you want to overwrite is contained in another application.

Never deploy an assembly from Visual Studio on a production computer

During the development process, the developer often must redeploy assemblies from Visual Studio. To enable the redeployment, Visual Studio may undeploy, unbind, stop, and unenlist artifacts included in the assemblies. Although this is necessary and appropriate in the development environment, it can cause unexpected and undesired consequences in a production environment. For this reason, as well as to avoid the possibility of anyone's attempting to deploy an assembly from Visual Studio on a production computer, we recommend that you never install Visual Studio on a production computer.

In addition, never refer to a production database from a computer running Visual Studio.

BizTalk Application Deployment and Management Checklists

This section provides checklists for the basic deployment and maintenance tasks that you may want to perform for BizTalk applications. Each checklist provides a list of the subtasks involved and links to the topics that describe how to perform them.

In This Section

- Checklist: Deploy a BizTalk Application
- Checklist: Update and Redeploy a BizTalk Application
- Checklist: Undeploy a BizTalk Application

Checklist: Deploy a BizTalk Application

Step	Reference
Learn about BizTalk application deployment features and functionality. You might also want to consider how to use binding files to make application deployment quicker and easier.	Understanding BizTalk Application Deployment and Management, Binding Files and Application Deployment
Ensure that you have appropriate permissions to perform the deployment.	Permissions Required for Deploying and Managing a BizTalk Application
In Visual Studio 2005, set deployment properties for each project in the BizTalk solution. Properties include the name of the database server and BizTalk Management database for the group, and the name of the destination BizTalk application. You can also enable redeployment and specify the option to automatically restart the host instances upon redeployment.	How to Set Deployment Properties in Visual Studio
Deploy the BizTalk assemblies from Visual Studio into the destination application.	How to Deploy a BizTalk Assembly from Visual Studio
Complete the BizTalk application, such as by adding .NET assemblies, Readme files, environment-specific binding files, and scripts; and configuring ports and receive locations. Also add any infrastructure, such as file drop locations, to the host computer.	Creating and Modifying BizTalk Applications, Binding Files and Application Deployment
Export the BizTalk application to an .msi file that you will use to import the application into another BizTalk group as well as install the application onto the computers that will run it.	How to Export a BizTalk Application
Import the .msi file to create the BizTalk application in the destination BizTalk group. If you have added environment-specific binding files to the application, you can select the bindings to apply on import.	How to Import a BizTalk Application
Make any additional modifications to the application so it runs in the destination environment, such as changing port configurations. If you have made any changes, export the application into an .msi file that you will use to install the application on the computers that will run it. You can also import the application into other BizTalk groups where you	Creating and Modifying BizTalk Applications, How to Add a Binding File to an Application, How to Export a BizTalk Application, How to Import a BizTalk Application

want to deploy it.	
Using the .msi file, install the application on all of the computers that will run it, and start the application from the BizTalk Administration console.	How to Install a BizTalk Application

Checklist: Update and Redeploy a BizTalk Application

Follow the steps in this checklist if you want to change or update one or more artifacts in an application that has already been deployed, and then redeploy the application.

Step	Reference
Ensure that you have appropriate permissions to perform the deployment.	Permissions Required for Deploying and Managing a BizTalk Application
Make any necessary changes to your assemblies in Visual Studio. If you are updating an assembly that is running in a production environment, you should always increment the assembly version number. For background information, see Application Versioning. Then deploy the assemblies from Visual Studio into a BizTalk application in the development environment.	How to Deploy a BizTalk Assembly from Visual Studio
Test any new or changed artifacts, ensuring that any artifacts that may depend on the new or changed artifact are also tested. While testing, be sure to consider dependencies that may exist between this application and other applications.	**Testing BizTalk Applications**
Add, remove, or reconfigure artifacts in the application as necessary.	How to Create or Add an Artifact, How to Remove an Artifact from an Application, Managing Artifacts
Export the application containing the new or changed artifacts into an .msi file. You may either export all of the artifacts in the application or only the artifacts you want to add or update. Be aware that file artifacts can be overwritten. If you export a file artifact, make sure that it is the version that you want to use in the application.	Exporting BizTalk Applications, Bindings, and Policies
If the update will interfere with the application as it runs, stop the application that you want to update. If you are updating an assembly with a new version, you do not need	How to Start and Stop a BizTalk Application

to restart the application.	
Import the changed or updated artifacts from the .msi file into the application that you want to update. Be sure to specify the option to overwrite existing artifacts.	How to Import a BizTalk Application
Install the updated application or artifacts from the .msi file onto all of the computers that run the application as well as any computers running applications that depend on this application.	How to Install a BizTalk Application
Start the application.	How to Start and Stop a BizTalk Application
After importing an assembly that contains an orchestration, if the application to which you are importing it already contains an assembly that has the same name, public key token, and version, stop and start the host instances of the host to which the orchestration is bound. This ensures that the new version of the assembly is used by BizTalk Server.	How to Stop a Host Instance, How to Start a Host Instance

Checklist: Undeploy a BizTalk Application

Step	Reference
Delete the BizTalk application from the BizTalk group. This removes it from the BizTalk Administration console and the BizTalk Server databases.	How to Delete a BizTalk Application from the BizTalk Group
Uninstall the application from any computers on which it has been installed.	How to Uninstall a BizTalk Application
If necessary, remove the assemblies that were included in the application from the global assembly cache (GAC).	How to Uninstall an Assembly from the GAC
Remove other files and settings from the local computer that were not automatically removed in the previous steps.	How to Remove Other Files and Settings for a BizTalk Application

BizTalk Application Deployment and Management Walkthroughs

This section provides walkthroughs that will help you learn how to use the new application deployment features of BizTalk Server 2006. A walkthrough provides step-by-step instructions for performing the subtasks involved in a more global task. For example, the first walkthrough provides step-by-step instructions for performing all of the tasks involved in deploying a BizTalk application, starting with deploying it to a test environment from a development computer, and then covering the tasks involved in deploying the application from test to staging and then production servers.

In This Section

- Walkthrough: Deploying a Basic BizTalk Application
- Walkthrough: Distributing Application Artifacts Among Multiple Host Instances
- Walkthrough: Updating the Artifacts in a Deployed BizTalk Application

Walkthrough: Deploying a Basic BizTalk Application

Microsoft BizTalk Server 2006 includes new features that simplify managing and deploying BizTalk business solutions. It now provides a BizTalk application container for the items in a business solution, such as orchestrations, schemas, maps, pipelines, schemas, and .NET assemblies. You can manage, modify, deploy, and install all of the items in an application as a single unit. BizTalk Server 2006 also includes new wizards to help automate application deployment tasks. For background information, see New Application Deployment and Management Features and Application Deployment and Management Tools.

This walkthrough provides step-by-step instructions for using the new deployment features. You will create a simple BizTalk application and then deploy it from the development environment to a test environment, and from there to staging and production environments. After completing this walkthrough, you will understand how to perform the following tasks:

- From within Visual Studio 2005 on a development computer, use the Deploy command to deploy BizTalk assemblies to a local instance of BizTalk Server. This creates a BizTalk application that is populated with the assemblies. A BizTalk assembly contains resource information such as orchestrations, pipelines, schemas, and maps to be used in a BizTalk solution.
- From the BizTalk Server Administration console, add and remove any items (called *artifacts*) as needed to create a fully functional business solution. Artifacts include orchestrations, send and receive ports, policies, maps, schemas, pipelines, assemblies, and scripts.
- Use the new Export, Import, and Installation Wizards to deploy the BizTalk application to a test computer for functional and system testing.
- Use the new Export, Import, and Installation Wizards to deploy the application to a staging server for final configuration and then deploy it to a production server.

Prerequisites

You have two options for setting up your beta test environment for this walkthrough:

- You can perform the tasks in this walkthrough on a single computer.
- You can more closely simulate a real-world deployment by setting up different computers to use as development, test, staging, and production computers. None of the tasks in this walkthrough should be performed in an actual production environment, however.

To perform the steps in this walkthrough, make sure that your beta test environment meets the following prerequisites:

- The development computer from which you deploy the BizTalk assemblies has Microsoft Visual Studio 2005 installed.
- Each computer used in the application deployment process described in this walkthrough, including the development computer, has BizTalk Server 2006 installed.
- Each BizTalk Server instance is a separate installation; in other words, it has its own BizTalk Server databases and groups.

In addition to the preceding requirements, you need to have available a Visual Studio solution or project that contains BizTalk assemblies. If you don't have an existing solution or project, you can use the ErrorHandling sample solution included with the BizTalk Server 2006 SDK for this purpose. Instructions for using this sample are included later in this walkthrough.

You also must have a user account that is a member of the BizTalk Server Administrators group and the local Administrators group on the computers you will be using to perform the tasks in this walkthrough.

Assumptions

This walkthrough assumes the following:

- You have a basic knowledge of BizTalk Server 2006. For more information, see Getting Started .
- The BizTalk assemblies that you are using have not been previously deployed to an application in your Beta test environment. If they have, you should undeploy the BizTalk application to which they were deployed. For instructions, see Undeploying BizTalk Applications.
- No application resources are shared with other applications.

Audience

The audience for this walkthrough is:

- **BizTalk application developers.** Developers can learn how to set project properties in Visual Studio 2005 and deploy BizTalk assemblies from Visual Studio to a BizTalk application. Developers also learn how to add artifacts to the application and then export the application to an .msi file. For background information about the application deployment tasks for a developer, see Development Tasks for BizTalk Application Deployment.

- **BizTalk application testers.** Testers can learn how to import the .msi file into BizTalk Server running on their test computer, which registers it as a BizTalk application. Testers can then learn how to install the application on the test computer and verify the installation. For background information about the application deployment tasks for testing, see Testing Tasks for BizTalk Application Deployment.

- **BizTalk Server IT administrators.** IT administrators who are responsible for deploying BizTalk applications to staging and production servers can learn the basic steps required for this task. For background information about the application deployment tasks for an IT administrator, see Staging Tasks for BizTalk Application Deployment and Production Tasks for BizTalk Application Deployment.

Overview of this walkthrough

The purpose of this walkthrough is to deploy a BizTalk application in a lab environment to assess how this technology would function if deployed in your production environment. The simple scenario covered in this walkthrough – deploying a single .msi file as a BizTalk application on a single computer – will help you become familiar with the basic tasks involved in application deployment.

The instructions provided in this document cover the following tasks:

1. **Configuring required permissions.** Before you begin the walkthrough, you need to ensure that you have the appropriate permissions to perform each task.

2. **Deploying BizTalk assemblies from Visual Studio.** This step is taken by the application developer. Deploying BizTalk assemblies from within Visual Studio automatically builds the assemblies and deploys their contents into a BizTalk application. If the application does not already exist, deploying the assemblies also creates the application. The application's artifacts are registered and their data stored in the BizTalk Management database. After the application has been created, you can view and manage its artifacts from within the BizTalk Server Administration Console. In the Administration Console, each application is stored in its own folder, with subfolders containing references to all of the application's artifacts.

3. **Modifying the BizTalk application by adding and removing artifacts.** This step can be taken by the application developer or IT administrator to add any artifacts that are required for the application to function correctly. From within the Administration Console, you can easily modify an application, for example, by adding and removing artifacts such as send and receive ports, scripts, and

additional assemblies. After you are satisfied that the application contains the required artifacts, you export it into an .msi file, as described next.

4. **Exporting the application into an .msi file.** This step can be taken by the developer, tester, or IT administrator to generate an .msi file that can be used to deploy a BizTalk application onto a different server. For example, the developer can export an .msi file that the tester can use to deploy the application onto a test server. After testing is complete, the tester can export an .msi file that the IT administrator can then use to deploy the application to a staging or production server (as described next). This walkthrough covers how to export an application into an .msi file by using the Export Wizard, which is available from the BizTalk Server Administration Console.

5. **Importing the .msi file and installing the application.** This step can be taken by the tester or IT administrator to deploy a BizTalk application onto a staging or production server. For example, the tester can import the application from an .msi file provided by the developer into a BizTalk group on a test computer and then install the application from the .msi file in order to test it. The IT administrator can likewise deploy the application from an .msi file provided by the tester onto a staging or production server. This walkthrough covers how to use the Import Wizard to import the .msi file into an application in a BizTalk group. As in Step 2, the application's artifacts are registered and their data stored in the BizTalk Server databases. This walkthrough also covers how to install the application on the current server by using the Installation Wizard or by simply double-clicking the .msi file. This allows you to run the application on the current server.

Step-by-step guide to deploying a BizTalk application

This section provides step-by-step procedures to deploy a BizTalk application through all of the phases, from development to test to staging to production. As previously mentioned, you can perform these steps all on the same computer, or if you want to more closely simulate your environment, you can use multiple computers.

1. Configure permissions

The first step is to ensure that you have the appropriate permissions to perform the tasks in this walkthrough. For more information, see Permissions Required for Deploying and Managing a BizTalk Application.

2. Deploy the BizTalk Assemblies

From within Microsoft Visual Studio 2005 on the development computer, use the procedures in this step to deploy BizTalk assemblies into a BizTalk application.

Before you begin, you must have a BizTalk solution available in Visual Studio. You can either create your own solution or project, or you can set up the ErrorHandling sample included with this Beta release. You can set up the ErrorHandling sample solution in Visual Studio as follows.

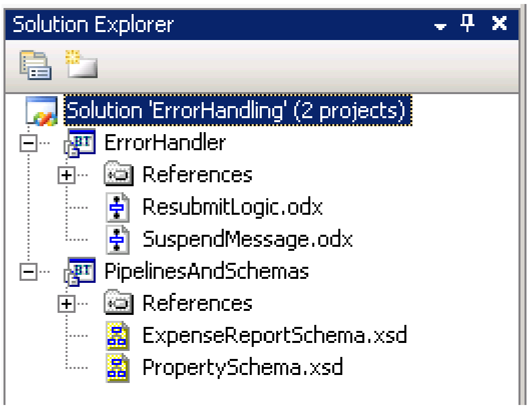
To set up the ErrorHandling solution

1. On the development computer, navigate to:

   %systemdrive%\Program Files\Microsoft
   BizTalk Server 2006\SDK\Samples\Messaging\ErrorHandling\ErrorHandler

2. Double-click **ErrorHandler.btproj**.

   The ErrorHandler solution opens in Visual Studio. This solution consists of two projects: ErrorHandler and PipelinesAndSchemas.



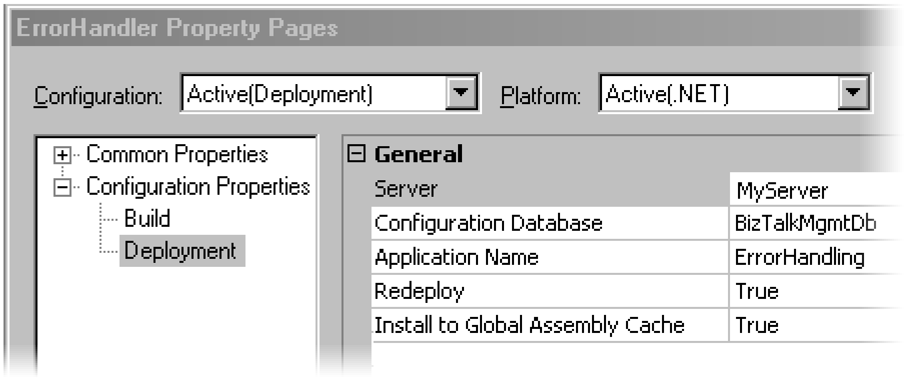
Next, you must set properties for each project in the solution. The ErrorHandling sample solution includes two projects for which you should set properties: ErrorHandler and PipeLinesAndSchemas. Configure properties to reflect the environment of the development computer. For example, the SQL server that you specify should be the instance running on the development computer, and which is hosting the local BizTalk Management database.

To set project properties

1. In Visual Studio, open the BizTalk solution.

2. In **Solution Explorer** right-click the project for which you want to configure properties, and then click **Properties**.

3. In the **Project Properties** tree view, expand **Configuration Properties**, and then click **Deployment**.



Configuration: Active(Deployment) Platform: Active(.NET)

General

Server	MyServer
Configuration Database	BizTalkMgmtDb
Application Name	ErrorHandling
Redeploy	True
Install to Global Assembly Cache	True

4. Configure project properties as described in the following table, and then click **OK**.

5. Repeat steps 1, 2, and 3 for each project in the solution.

Property	**Setting**	**Explanation**
Server	<Server name>	Name of the SQL Server instance that hosts the BizTalk Management database on the local computer. This is usually the name of the local computer.
BizTalk Management database	<BizTalk Management database name>	Name of the BizTalk Management database on the local computer. The default name is BizTalkMgmtDb.
Application Name	<Name>	Name of the BizTalk application to which to deploy this project. If the application exists, the project assembly will be added to it. If the application does not exist, it will be created. If this field is blank, the assemblies will be deployed to the default BizTalk application. For the purposes of this walkthrough, specify the same application name for each project (ErrorHandling, if you use the sample BizTalk project) so that all assemblies will be deployed into the same application.
Redeploy	True	Setting this to True allows you to redeploy the assemblies without changing their version number.
Install to Global Assembly Cache	True	Setting this to true deploys the assemblies to the BizTalk Management database. Set this to false only if you plan to use the global assembly cache (GAC) with other tools, such as gacutil.

In the process of deploying BizTalk assemblies, Visual Studio first builds them. The build process requires that each project is associated with a strong name assembly key file. If you haven't already done so, use the following procedure to generate a strong name assembly key file.

To create a strong name assembly key file

1. On the **Start** menu, point to **Programs**, point to **Microsoft Visual Studio 2005**, point to **Visual Studio Tools**, and then click **Visual Studio 2005 Command Prompt**.

2. At the command prompt, from the folder where you want to store the key file, type the following command, and then press ENTER:

   sn -k *file_name***.snk**

   Example: **sn -k ErrorHandling.snk**

   A confirmation message, **Key pair written to <resource_file>.snk**, displays on the command line.

Next you need to associate each project with the key file.

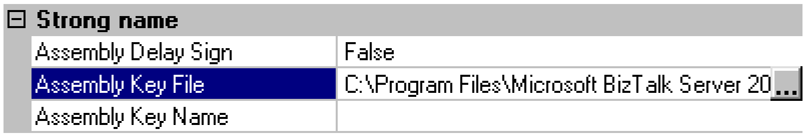
To associate your projects with the key file

1. In Visual Studio Solution Explorer, right-click the project and then click **Properties**.

2. Expand Common Properties, and then click **Assembly**.

3. In the right pane, scroll to **Strong name**.

4. Click the field next to **Assembly Key File**, click the ellipsis (**...**) button**,** and then browse to the key file.



⊟ Strong name	
Assembly Delay Sign	False
Assembly Key File	C:\Program Files\Microsoft BizTalk Server 20 ...
Assembly Key Name	

5. Click the key file, click **Open**, and then click **OK**.

6. Repeat steps 1 through 5 for each project in your solution.

Now you can build and deploy all of the assemblies in the solution in one step, as follows.

To deploy the assemblies in a solution

- In Visual Studio **Solution Explorer**, right-click the solution, and click **Deploy Solution**.

The status of the build and deployment process displays in the lower left corner of the page. If you are using the ErrorHandling sample solution, several warning messages will display. You can ignore these. When the deployment is complete, "Deploy succeeded" displays.

Deploying BizTalk assemblies registers them as part of the specified BizTalk application in the BizTalk Management database. It also populates the database with all of the items, or *artifacts*, contained in the assemblies. If the application did not already exist before the deployment, this step also creates a new application. You can now view the BizTalk application and its artifacts from within the BizTalk Server Administration Console on the development computer.

To view the BizTalk application and its artifacts

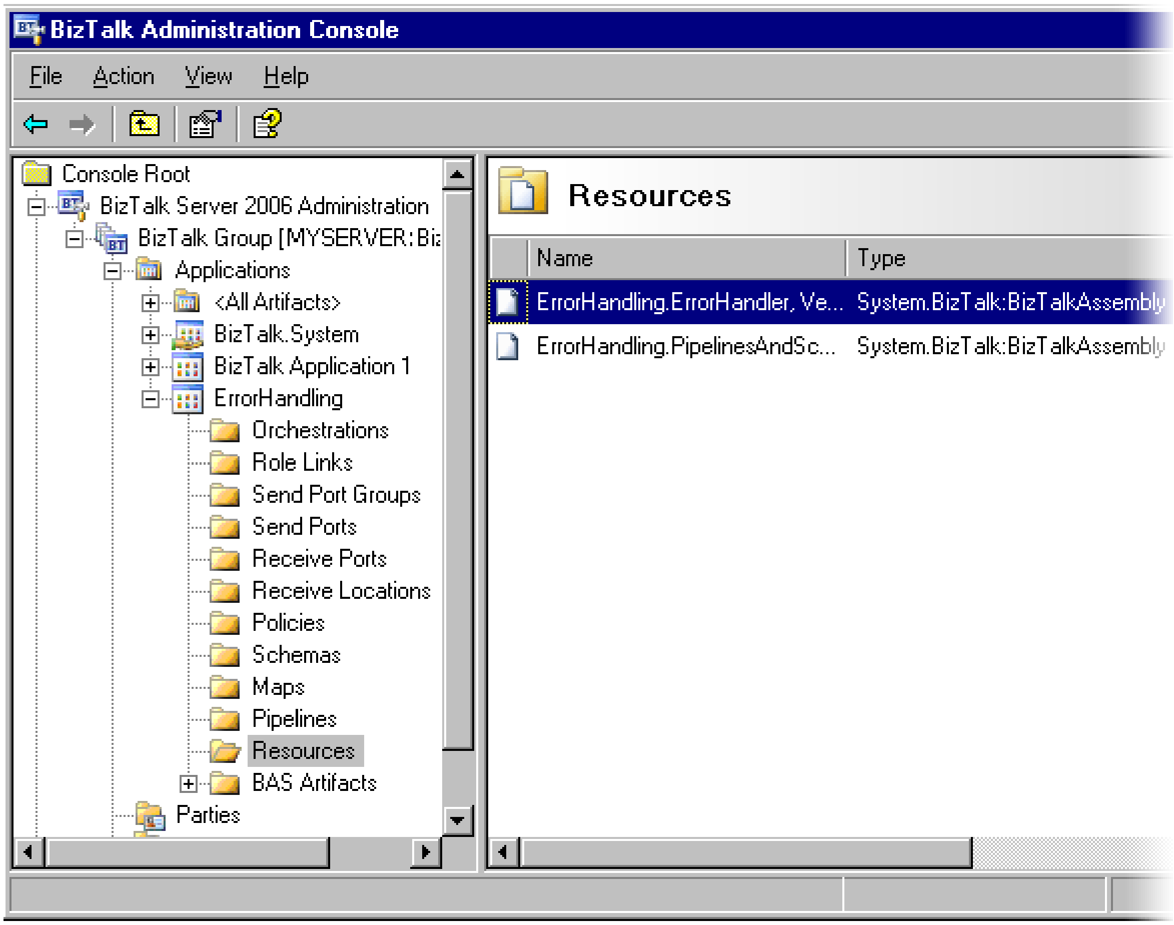
1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

2. In the console tree, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, and then expand **Applications**.

3. Expand the folder of the application to which you just deployed your assemblies.

4. Click the folders under the application folder to view their contents.

The following screen shows an application in the BizTalk Server Administration Console.



Name	Type
ErrorHandling.ErrorHandler, Ve...	System.BizTalk:BizTalkAssembly
ErrorHandling.PipelinesAndSc...	System.BizTalk:BizTalkAssembly

In the appropriate folders, you should see the artifacts that were contained in the assemblies that you deployed. If you deployed the ErrorHandling sample BizTalk Solution, you should see artifacts in the Orchestrations, Schemas, and Resources folders.

3. Add and configure artifacts

From the BizTalk Server Administration Console, you can modify your application by adding and configuring artifacts. For example, you might want to configure send and receive ports or add pre-processing scripts. You can also remove artifacts from the application. You can try out this functionality using the following procedure. The ErrorHandling sample does not include additional artifacts that you can add; however, you can test this functionality by adding items that might already exist in your environment. For more information about modifying your application, see Creating and Modifying BizTalk Applications.

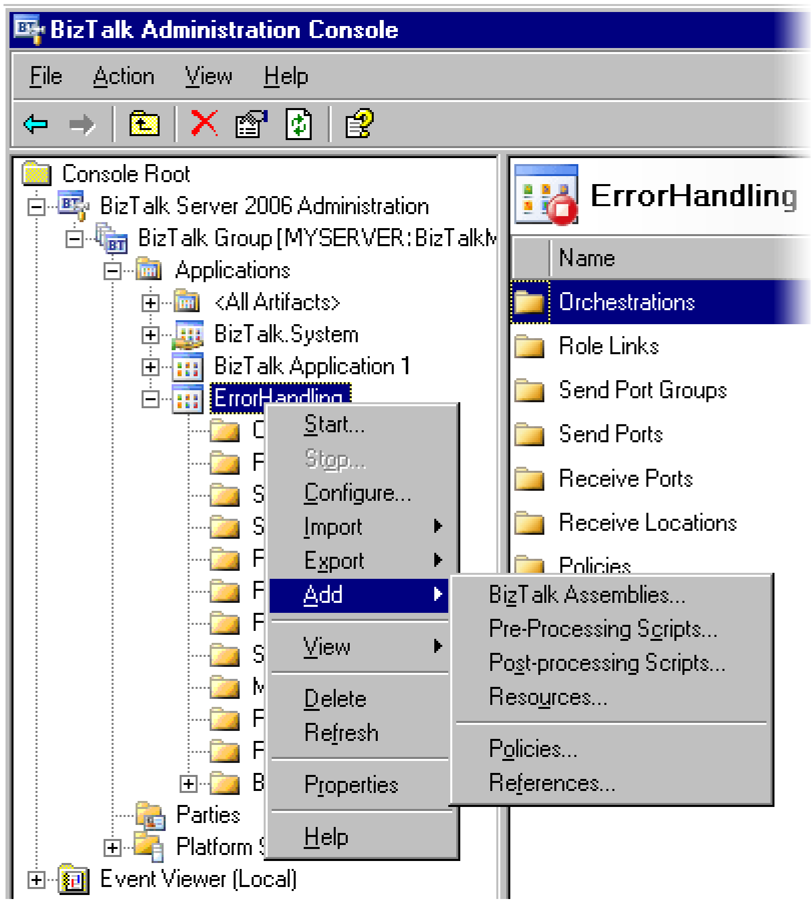
To add an artifact to an application

1. If necessary, open the BizTalk Server Administration Console as follows: 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 **BizTalk Group**, and then expand **Applications**.

3. To add the following types of artifacts, right-click the ErrorHandling application folder, and then click **Add**.

- BizTalk assemblies
- Pre-processing scripts
- Postprocessing scripts
- Resources
- Policies
- References



4. To add the following artifacts, expand the application folder, right-click the folder for the type of artifact to add, and then click **Add**.

- Send ports groups
- Send ports
- Receive ports
- Receive locations
- Policies
- Resources, including BizTalk assemblies, preprocessing scripts, post-processing scripts, and other resources.

5. To add BAS artifacts, expand the application folder, expand the BAS node, right-click the artifact folder, and then click **Add**.

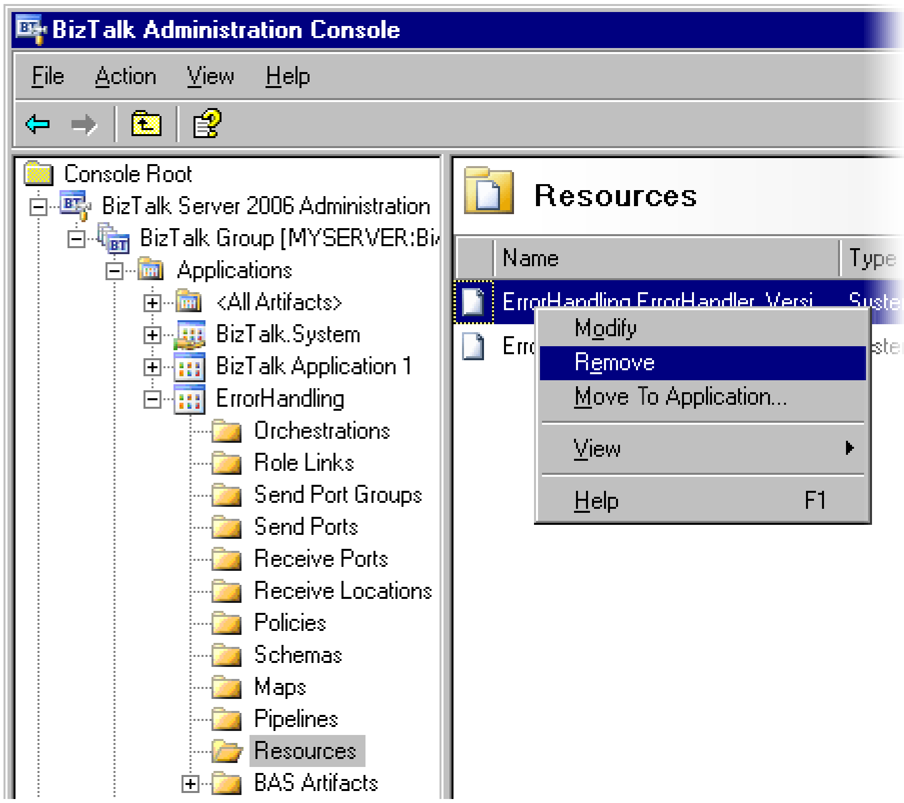
You can also remove an artifact from an application.

To remove an artifact from an application

1. If necessary, open the BizTalk Server Administration Console as follows: 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 **BizTalk Group**, and then expand **Applications**.

3. Expand the folder containing the artifact, right-click the artifact, and then click **Remove**.



4. Export the application

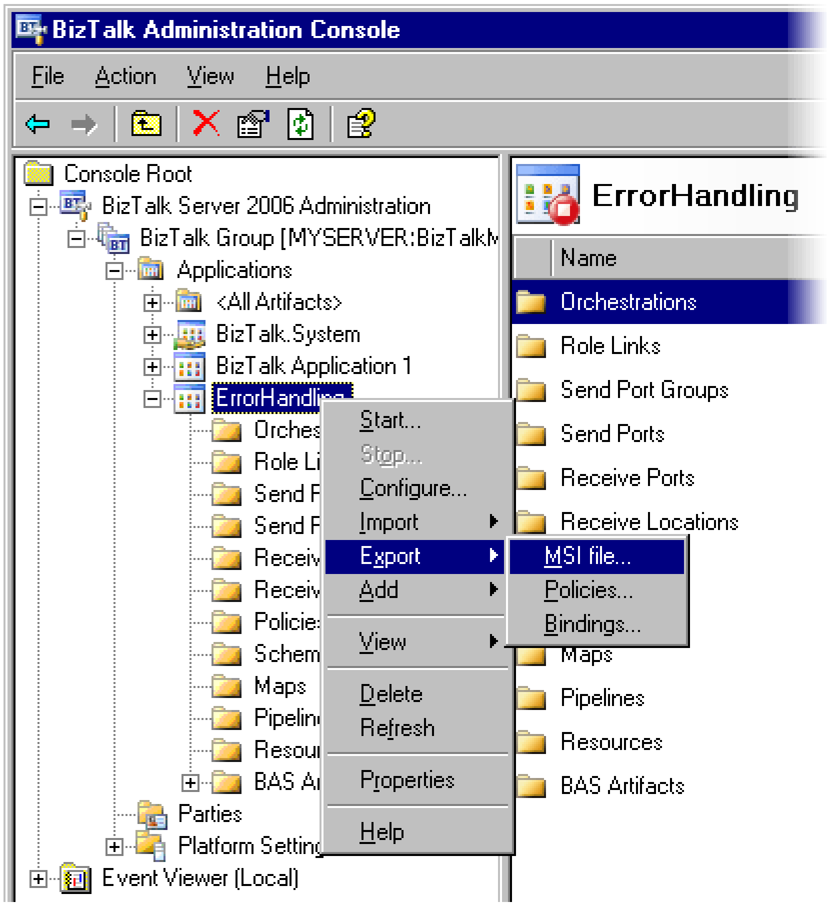
After you have created a BizTalk application and modified it as necessary, you can export the application from within the BizTalk Server Administration Console. This generates an .msi file that you can later import into another BizTalk group to recreate the application in the new group. To run the application on a particular server, you must also install it locally from the .msi file.

To export the application

1. If necessary, open the BizTalk Server Administration Console as follows: 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 **BizTalk Group**, and then expand **Applications**.

3. Right-click the BizTalk application, point to **Export**, and then click **MSI file**.



4. On the Welcome to the Export Wizard page, click **Next**.

5. On the Select Resources page, select the resources to export into the .msi file, and then click **Next**. For this walkthrough, you can accept the defaults.

6. If the Specify IIS Hosts page displays, specify the host name and protocol of any virtual directories included in your application, and then click **Next**. This page displays only if the application has not previously been imported into this instance of BizTalk Server.

7. On the Dependencies page, review the dependencies for the application, and then click **Next**. This walkthrough assumes that the application has no dependencies.

8. On the Destination page, in **Destination application name**, type the application name.

9. In **MSI file to generate**, type the full path for the .msi file, and then click **Export**. You might want to type a short path to make it easier to locate the .msi file on your hard drive. Example: C:\.msi\Errorhandling.msi

10. On the Summary page, make a note of the location of the log file for this operation, and then click **Finish**.

In the file system, verify that the .msi file was created in the location that you specified.

5. Import and install the application

The next step is to import the application into a BizTalk Server group and then install the application. You import the application by selecting the .msi file that you generated in the previous step from within the Import Wizard.

You can repeat the tasks in this step each time you want to migrate an application from one instance of BizTalk Server to another, such as when migrating from a development computer to a test computer, a test computer to a staging computer, or a staging computer to a production computer.

At this point, if you are using only one computer for this walkthrough, you might want to delete the application from the BizTalk Server Administration Console. You might also want to delete the assemblies from the global assembly cache (GAC). This way, when you import the application, you will be able to verify that it has been re-created correctly. If you are using multiple computers for this walkthrough, you do not need to perform these tasks.

To delete the application from the BizTalk group

1. If necessary, open the BizTalk Server Administration Console as follows: Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

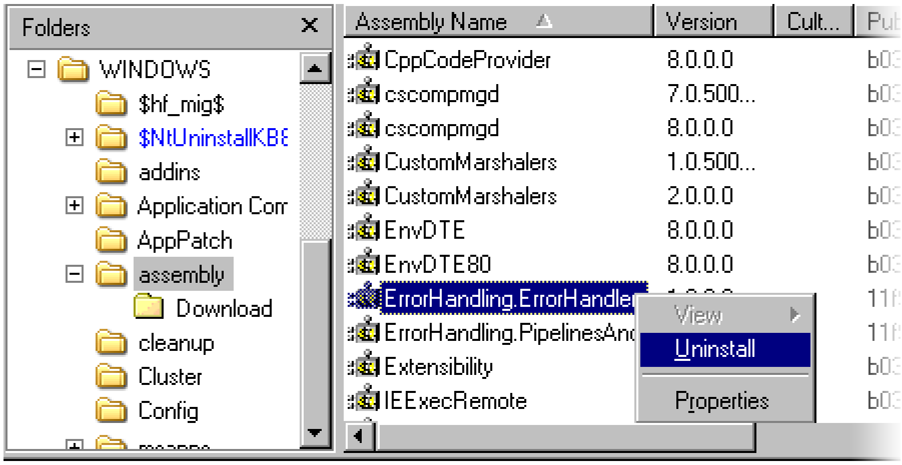
2. Right-click the application and click **Delete**.

To delete assemblies from the global assembly cache

1. In the file system, navigate to %systemdrive%\Windows\Assembly.

2. Right-click each assembly file that was generated for your solution, click **Uninstall**, and then click **Yes** to confirm. For example, the assembly files

associated with the ErrorHandling project are ErrorHandling.ErrorHandler and ErrorHandling.PipelinesAndSchemas.



Assembly Name	Version	Cult...	Pul
CppCodeProvider	8.0.0.0		b03
cscompmgd	7.0.500...		b03
cscompmgd	8.0.0.0		b03
CustomMarshalers	1.0.500...		b03
CustomMarshalers	2.0.0.0		b03
EnvDTE	8.0.0.0		b03
EnvDTE80	8.0.0.0		b03
ErrorHandling.ErrorHandler			11f
ErrorHandling.PipelinesAnd			11f
Extensibility			b03
IEExecRemote			b03

Now you are ready to import the application into a BizTalk group. If you want to import the application into a BizTalk group running on another computer, you must first copy the .msi file to the other computer before you can import and install the application. If you are performing this walkthrough on a single computer, you do not need to perform this extra task; the .msi file is already present on the local computer.

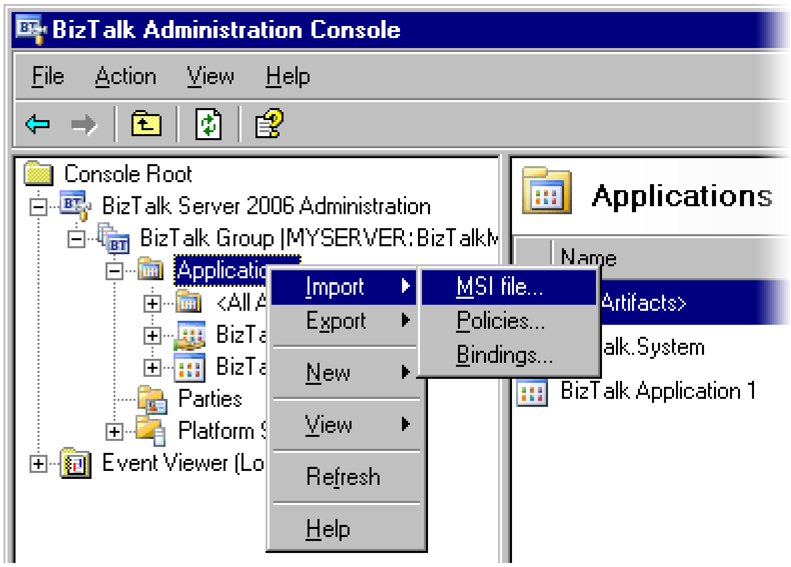
You can optionally install the application as soon as you import it by using the Installation Wizard, which is the approach we will take in this walkthrough. If you prefer to install the application at a different time, you can double-click the .msi file in the file system. Installing the application adds it to the global assembly cache (GAC), which enables the application to run on the local computer.

To import and install the application

1. Open the BizTalk Server Administration Console for the instance of BizTalk Server into which you want to import the application as follows: Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

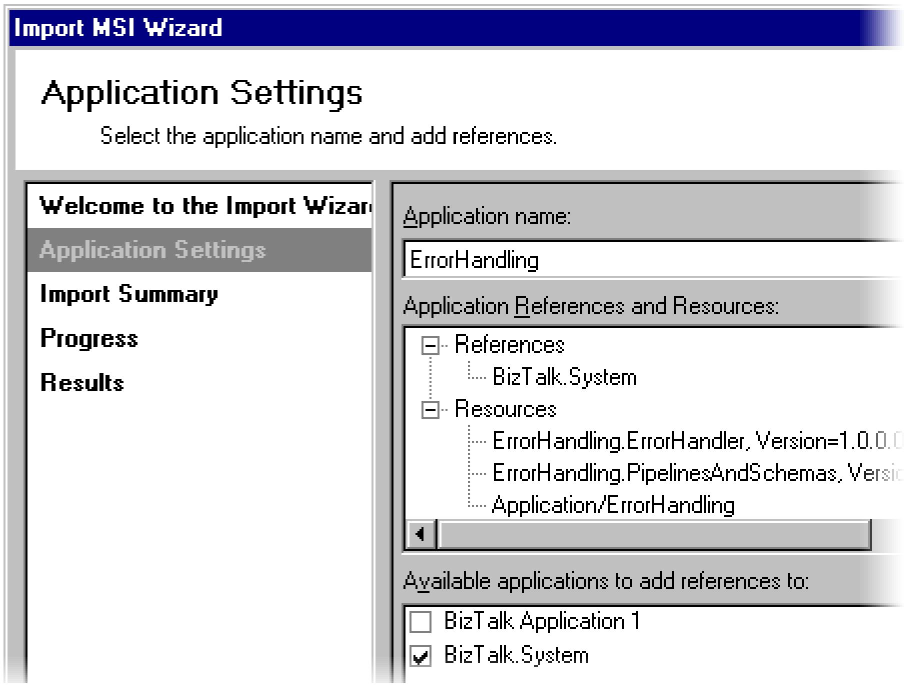
2. In the console tree, expand **BizTalk Server 2006 Administration**, and then expand **BizTalk Group**.

3. Right-click **Applications**, point to **Import**, and then click **.msi file**.



4. On the Welcome to the Import Wizard page, in **.msi file to import**, type the complete path of the .msi file, and then click **Next**. Example: C:\.msi\MyApplication.msi

5. On the Application Settings page, in **Available applications to add references to**, select the applications to which to add references, and then click **Next**. If you are using the ErrorHandling sample solution, you can accept the defaults.


Import MSI Wizard

Application Settings

Select the application name and add references.

Application name:

ErrorHandling

Application References and Resources:

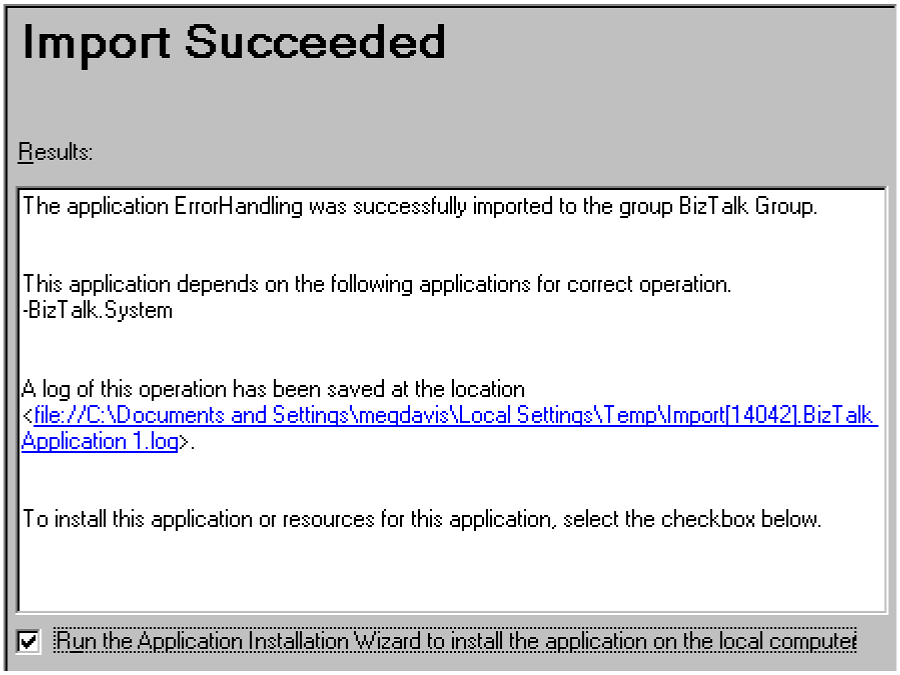
- References
  - BizTalk.System
- Resources
  - ErrorHandling.ErrorHandler, Version=1.0.0.0
  - ErrorHandling.PipelinesAndSchemas, Versio
  - Application/ErrorHandling

Available applications to add references to:

□ BizTalk Application 1
☑ BizTalk.System

6. On the Import Summary page, confirm that the summary information is correct, and then click **Import**.

7. On the final screen of the Import .msi Wizard, select **Run the Application Installation Wizard to install the application on the local computer**, and then click **Finish**.



Import Succeeded

Results:

The application ErrorHandling was successfully imported to the group BizTalk Group.

This application depends on the following applications for correct operation.
-BizTalk.System

A log of this operation has been saved at the location
<file://C:\Documents and Settings\megdavis\Local Settings\Temp\Import[14042].BizTalk Application 1.log>.

To install this application or resources for this application, select the checkbox below.

☑ Run the Application Installation Wizard to install the application on the local compute

8. On the Select Installation Folder page, in **Folder**, type the installation path for the BizTalk application, and then click **Next**.

9. Click **Next** on the next three pages to continue the installation.

   Windows Installer installs the application on the local computer.

10. On the Installation Complete page, click **Close**.

Next, you can make sure that the application was imported and installed by verifying the following:

- The application and all of its artifacts exist in the application's folder in the Administration Console.
- The application assemblies exist in the GAC.
- The files associated with the application exist in the path that you specified when you installed the application.

- If you have configured the application so that it can function, such as by specifying send and receive ports, you can now start the application. The ErrorHandling sample application is not set up to function by default, however, so unless you have manually configured it, you will not be able to start it.

To start the application

1. If necessary, open the BizTalk Server Administration Console as follows: 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 **BizTalk Group**, and then expand **Applications**.
3. Right-click the application, and then click **Start**.

If you want to move this application to another instance of BizTalk Server, you can follow the instructions in Steps 3, 4, and 5 of the **To import and install the application** section to do so. If you want to repeat these steps on the same computer, in addition to deleting the application and removing the assembly files from the GAC, you may also want to remove the BizTalk application from the computer before you begin. Then you later will be able to verify that it has been correctly reinstalled.

To remove the application from the computer

- In Add or Remove Programs, click the application, and then click **Remove**.

The application is uninstalled from the local computer.

Understanding BizTalk Application Deployment and Management

This section introduces the new application deployment and management features of BizTalk Server 2006 and provides background information that will help you understand how to use these features to deploy and manage BizTalk applications.

In This Section

- What Is a BizTalk Application?
- The Application Deployment Process
- New Application Deployment and Management Features
- Application Deployment and Management Tools
- Application Deployment and Management Scenarios

- What Happens to Artifacts During Application Deployment
- Application Deployment Tasks
- Security Considerations for Application Deployment
- Dependencies and Application Deployment
- Binding Files and Application Deployment
- Application Versioning
- Artifacts That Must Be Unique in an Application or Group
- Important Considerations for Updating Applications

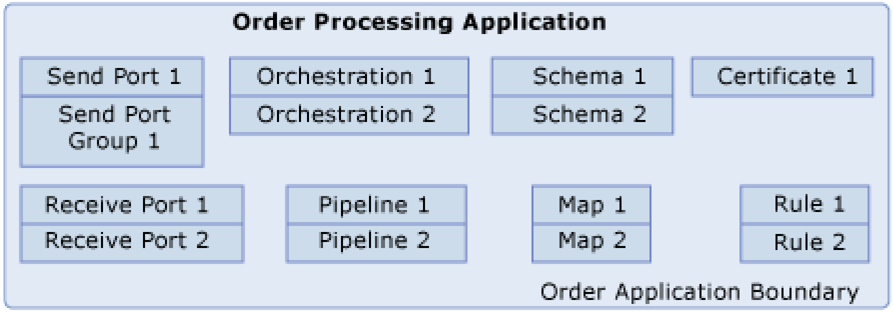
What Is a BizTalk Application?

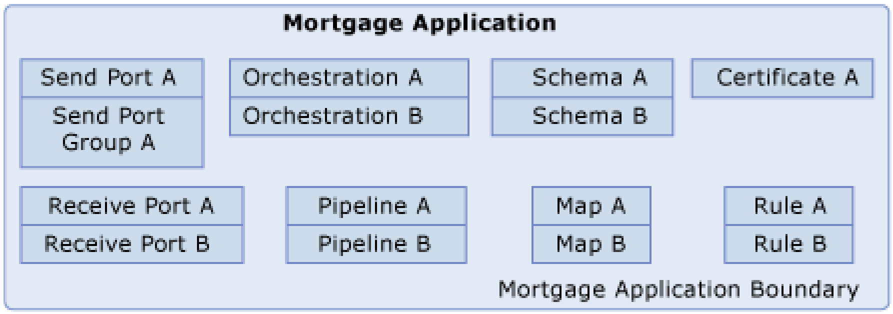
The BizTalk application is a new feature of BizTalk Server 2006 that makes it quicker and easier to deploy, manage, and troubleshoot BizTalk Server business solutions. A BizTalk application is a logical grouping of the items, called "artifacts," used in a BizTalk Server business solution.. Artifacts are described in more detail later in this topic.

The newly designed administration and monitoring tools of BizTalk Server 2006 take advantage of this new concept, so that you can manage and configure BizTalk Server business solutions at the application level, and not just at the individual artifact level. By creating an application and adding artifacts to it, you can view, package, deploy, and manage a group of artifacts in a solution as a single entity. Therefore, as the number of complex applications increases you can still manage them individually in a simple and intuitive manner.

There are several tools you can use to create and manage applications, which are described in Application Deployment and Management Tools.

The following diagram depicts two BizTalk applications and the artifacts that they contain.





Artifacts

Artifacts include the following:

- BizTalk assemblies and the BizTalk-specific resources that they contain – orchestrations, pipelines, schemas, and maps
- .NET assemblies that do not contain BizTalk-specific resources
- Policies
- Send ports, send port groups, receive locations, and receive ports
- Other items that are used by the solution, such as certificates, COM components, and scripts

For background information about each type of artifact, see Runtime Architecture. For more information about adding, removing, and configuring artifacts, see Managing Artifacts.

An application can contain all of the artifacts used in a business solution or only some of them. Depending on how you want to deploy the artifacts, you may want to place them into a single application or two or more applications. For more information about deciding how to group artifacts, see Best Practices for Deploying a BizTalk Application.

The default application

When BizTalk Server 2006 is configured following installation, a default application named BizTalk Application 1 is automatically created. This provides compatibility between BizTalk Server 2004 and BizTalk Server 2006. When you upgrade an instance of BizTalk Server 2004 to BizTalk Server 2006, the artifacts in the BizTalk Management database are automatically placed in the default application. If you want to manage your migrated business solutions as separate applications, you can create additional applications and move the artifacts that you want to manage together into them, as described in How to Move an Artifact to a Different Application. For information about best practices for grouping artifacts into different applications, see Best Practices for Deploying a BizTalk Application.You can also change the default application or rename the default application.

In the following scenarios, artifacts are automatically added to the default application:

- When you deploy an assembly from Visual Studio 2005 into BizTalk Server without specifying an application name. For more information, see How to Deploy a BizTalk Assembly from Visual Studio.

- When you use the BTSDeploy command-line tool to add an artifact to an application. (Because it was developed for previous versions of BizTalk Server that did not include the application feature, BTSDeploy does not provide an option to specify an application name.) For more information, see Deploying Assemblies Using BTSDeploy.

- When you use BTSTask to add an artifact to an application without specifying an application name. For more information, see AddResource Command.

- When you use BTSTask to import an application .msi file without specifying an application name. For more information, see ImportApp Command.

- When you configure BizTalk Server with Business Activity Services (BAS), the following artifacts are automatically created in the default application. You should not delete these artifacts because they are required for BAS to function. Any BAS artifacts that you add to an application in the BizTalk group will depend on them. For more information, see Managing BAS Artifacts.

    - A send port named Bas.TpmWebServicePort

    - A receive port named Sts.Outbox

    - A receive location named Sts.Outbox.Location

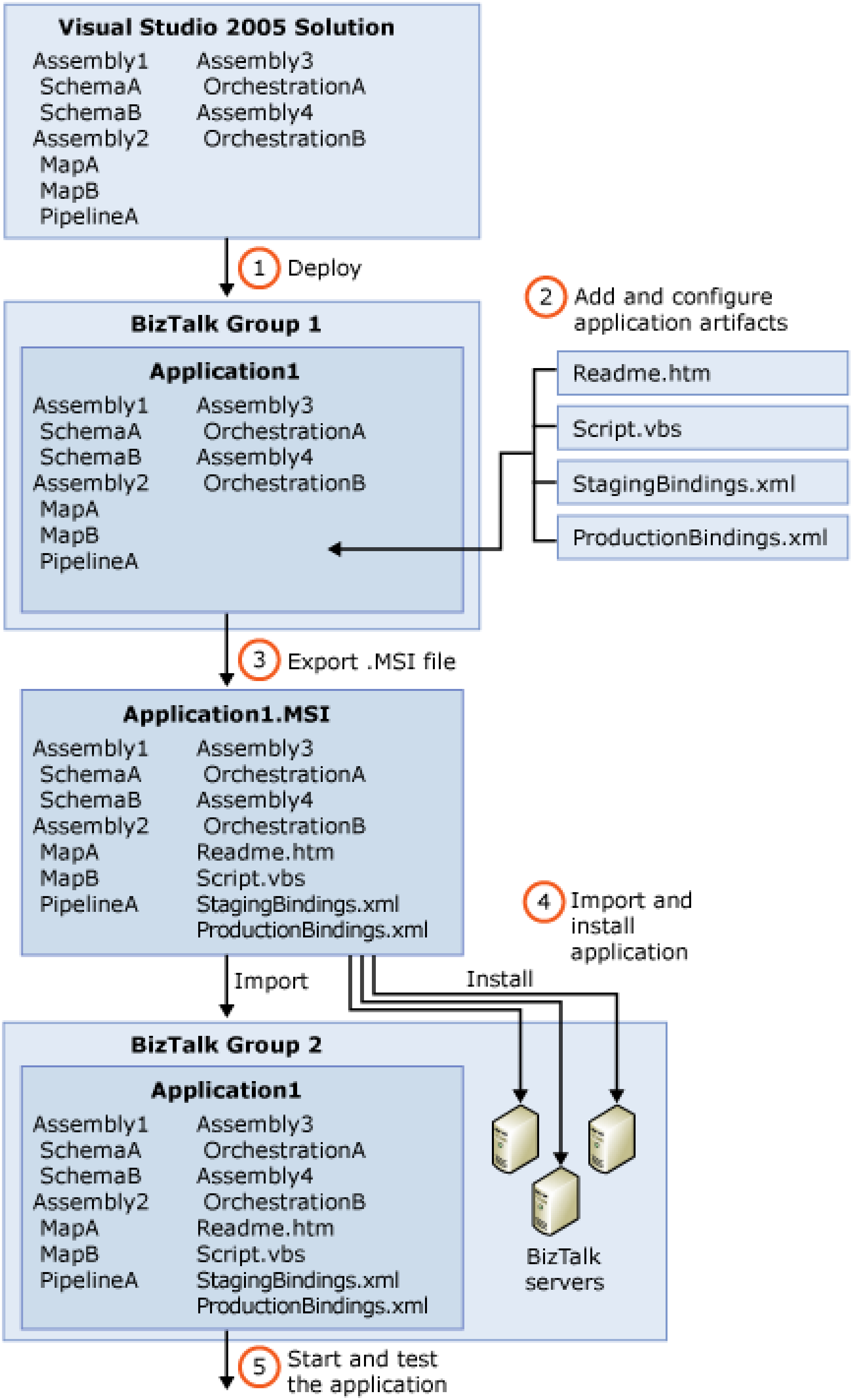
The BizTalk.System application

When BizTalk Server 2006 is configured following installation, an application named BizTalk.System is automatically created and populated with common artifacts that are used by all BizTalk applications, such as the default schemas and pipelines.

BizTalk.System and its artifacts are read-only. You cannot delete or rename BizTalk.System, nor can you cannot delete, rename, or move any of the artifacts that it contains.

The Application Deployment Process

This following diagram depicts the general steps involved in deploying a BizTalk application. For detailed information about the tasks involved during the development, testing, staging, and production phases of application deployment, see Application Deployment Tasks.



1. **Deploy from Visual Studio 2005 the assemblies in a BizTalk solution.** This builds the assemblies and imports them, along with the orchestrations, pipelines, schemas, and maps that they contain (called "artifacts") into the local BizTalk Management database. Deployment also associates them with the BizTalk application that you have specified in project properties within Visual Studio. For instructions, see Deploying BizTalk Assemblies from Visual Studio into a BizTalk

Application. After you deploy a solution, you can view and manage the deployed assemblies and their artifacts from within the BizTalk Server Administration console or by using the BTSTask command-line tool. You can manage the artifacts either individually or together, grouped within the application.

2. **Add and configure artifacts.** You can add artifacts, such as scripts and Readme files, to the application by using the Administration Console or BTSTask. You can also configure artifacts, such as send and receive ports, receive locations, and orchestrations by using the Administration Console. For more information, see How to Create or Add an Artifact and Managing Artifacts. You can also generate binding files and add them to the application if you want to apply different bindings for different environments into which you might import the application. For more information, see Binding Files and Application Deployment.

3. **Export the application into an .msi file.** You can use the Export MSI File Wizard or BTSTask to export the application artifacts into an .msi file that you will use to import the application into a new BizTalk group as well as install the application on the computers that will run it. For instructions, see How to Export a BizTalk Application.

4. **Import the application into another BizTalk group and install the application on the computers that will run it.** You can use the Import MSI Wizard or BTSTask to import the BizTalk application from the .msi file created in Step 3 into another BizTalk group to create the application and its artifacts in the new group. You then use the .msi file to install the application on the computers that will run it. You must do this before the application can run. If you are ready to test the application, you can import it into BizTalk group in a test environment and install it. If your application is ready for staging or production, you can import it into one of those environments and install it. For instructions, see How to Import a BizTalk Application and How to Install a BizTalk Application.

5. **Start the application and verify that it is functioning correctly.** You can start the application from the Administration Console, as described in How to Start and Stop a BizTalk Application. You can then test the application, as described in **Testing BizTalk Applications**.

New Application Deployment and Management Features

BizTalk Server 2006 includes new features that make deploying BizTalk business solutions quicker and easier than in previous releases of BizTalk Server.

- **BizTalk application.** A BizTalk application provides way to view and manage the items, called "artifacts," that comprise a BizTalk business solution. Artifacts include the assemblies, orchestrations, maps, schemas, security certificates, business rules policies, BAM configuration files, bindings, scripts, and so forth that are necessary for a business solution to function. You can add artifacts to a BizTalk application and then view, package, deploy, and manage all of an application's artifacts as a single entity from within the BizTalk Server Administration console. You can create one, two, or more BizTalk applications to manage the artifacts in your business solution. You can use the Export Wizard to export the application and its artifacts, and then use the Import Wizard to import the .msi into another BizTalk group to recreate it there. You can then use the Installation Wizard to install the application. These wizards are described later in this topic. Each deployment procedure in this documentation includes instructions for using the Administration Console. You can also use the BTSTask command-line tool for many administrative tasks, and this documentation includes procedures for using BTSTask as well.

- **Import Wizard.** You use the Import Wizard, available from the BizTalk Server Administration console, to import artifacts from an .msi file into a BizTalk application. If the specified application does not already exist, the Import Wizard creates it. The wizard also registers and stores the artifacts in the .msi file in the BizTalk Management database. For more instructions on using the Import Wizard to import artifacts, see Importing BizTalk Applications, Bindings, and Policies.

- **Installation Wizard.** You can start the Installation Wizard as a final step of the Import Wizard. This wizard installs the application on the local computer, so that you can run the application on the local computer. For instructions, see How to Import a BizTalk Application. You can also start the Import Wizard by double-clicking an application .msi file. For more information, see How to Install a BizTalk Application.

- **Export Wizard.** You use the Export Wizard, also available from the BizTalk Server Administration console, to generate an .msi file from a BizTalk application. You can then use the Import Wizard to import the .msi file into a different BizTalk group. This makes it easy to add resources to an application, or automatically create an application on the new BizTalk group. For more information about using the Export Wizard, see How to Export a BizTalk Application.

- **BTSTask command-line tool.** BTSTask supports the new application deployment features of BizTalk Server 2006, allowing you to perform many operations from the command line. For more information, see BTSTask Command-Line Reference.

To help you become familiar with these new features, we recommend that you perform the steps in Walkthrough: Deploying a Basic BizTalk Application.

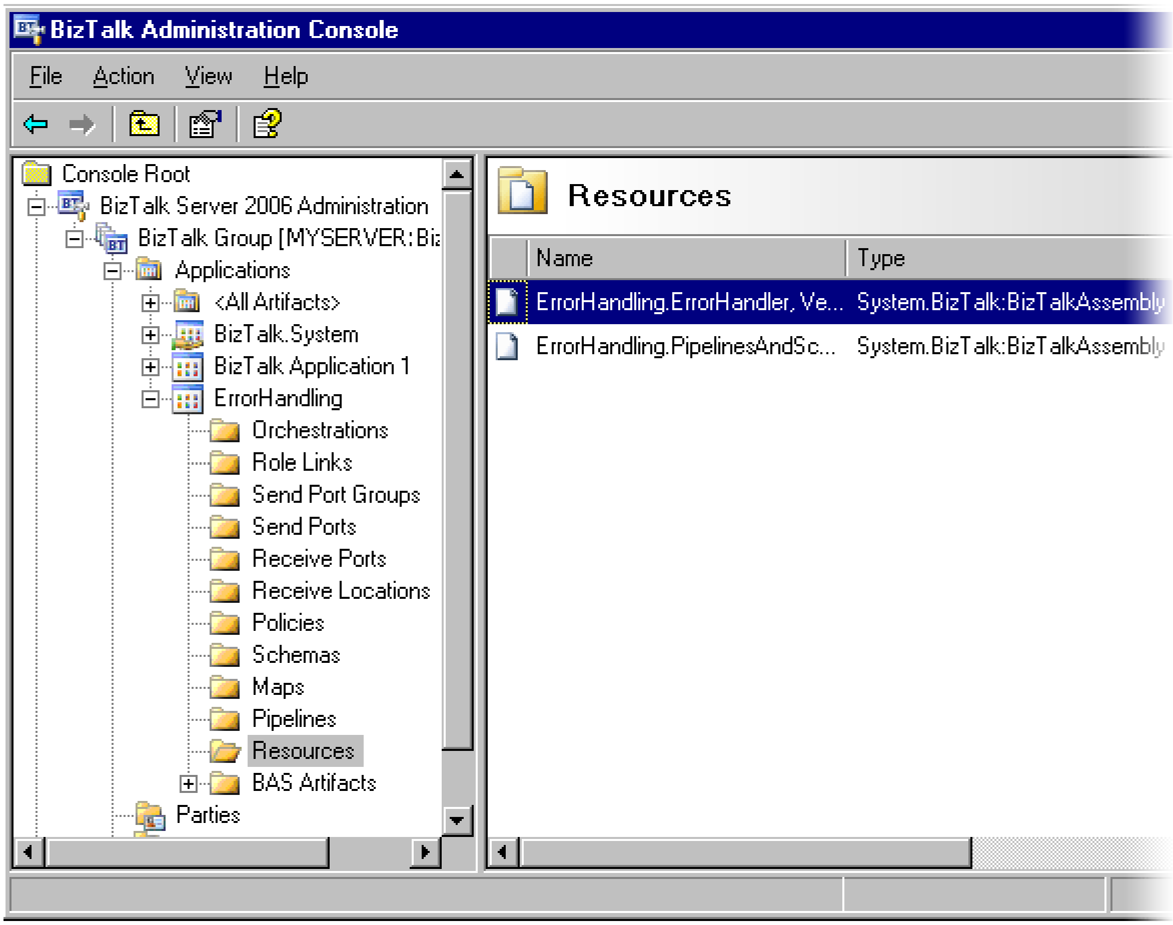
Application Deployment and Management Tools

This topic briefly describes the tools that you can use for deploying and managing a BizTalk application and then summarizes the operations you can perform with each tool. A table at the end of this topic summarizes the tasks you can perform using the BizTalk Server Administration console and BTSTask.

- **BizTalk Server Administration console.** The BizTalk Server Administration console, a Microsoft Management Console (MMC) snap-in, is the primary management tool for BizTalk Server. It provides a graphical user interface for performing all of the deployment and management operations for a BizTalk application. For example, from the administration console, you can start the

Import, Installation, and Export Wizards as well as add and remove an application's artifacts and make other modifications to the application. You also generate BizTalk .msi files for your BizTalk applications by using either the Export MSI File Wizard or BTSTask, described next. You can then install the application on a computer from the .msi file or import the application from the .msi file into another BizTalk group. For more information about the administration console, see Using the BizTalk Server Administration Console.

The following screen capture provides a view of a BizTalk application and its artifacts from within the BizTalk Server Administration console.



Name	Type
ErrorHandling.ErrorHandler, Ve...	System.BizTalk:BizTalkAssembly
ErrorHandling.PipelinesAndSc...	System.BizTalk:BizTalkAssembly

- **BTSTask command-line tool.** New with BizTalk Server 2006, BTSTask allows you to perform many application management tasks from the command line. For more information about using BTSTask, see BTSTask Command-Line Reference.

- **Scripting and programmability APIs**. You can use Microsoft Windows Management Instrumentation (WMI) or the BizTalk Explorer Object Model to create and run scripts that automate many application management tasks. For information about using WMI, see **WMI Class Reference**. For more information about the BizTalk Explorer Object Model, see Using the BizTalk Explorer Object Model from Managed Code

- **Visual Studio 2005.** The developer can create BizTalk assemblies in Visual Studio and use the Deploy command to automatically deploy them into a BizTalk application. For more information, see Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application. In addition, the developer can use BizTalk Explorer to configure application artifacts from within Visual Studio. For more information, see Managing Business Processes Using BizTalk Explorer .

The following table summarizes the tasks you can perform by using the administration console and BTSTask.

Task	**Tool**
Export an application .msi file	• BizTalk Server Administration console – Export MSI File Wizard • BTSTask
Import an application .msi file	• BizTalk Server Administration console – Import MSI Wizard • BTSTask
Create or delete an application	• BizTalk Server Administration console • BTSTask
Install an application	BizTalk Server Administration console – Installation Wizard (or double-click the application .msi file)
Uninstall an application	BTSTask (or use Add or Remove Programs Control Panel)
Add artifacts to an application or remove artifacts from an application	• BizTalk Server Administration console • BTSTask
Import and export bindings and binding files	• BizTalk Server Administration console • BTSTask
Import and export policies	• BizTalk Server Administration console • BTSTask
Start and stop an application	BizTalk Server Administration console

Change the state of an artifact: start, stop, enable, disable, enlist, and unenlist	BizTalk Server Administration console
Edit artifact properties	BizTalk Server Administration console
Create a send port, send port group, receive location, or receive port	BizTalk Server Administration console

Application Deployment and Management Scenarios

This section describes the various scenarios in which a BizTalk application might be deployed, updated, or undeployed.

In This Section

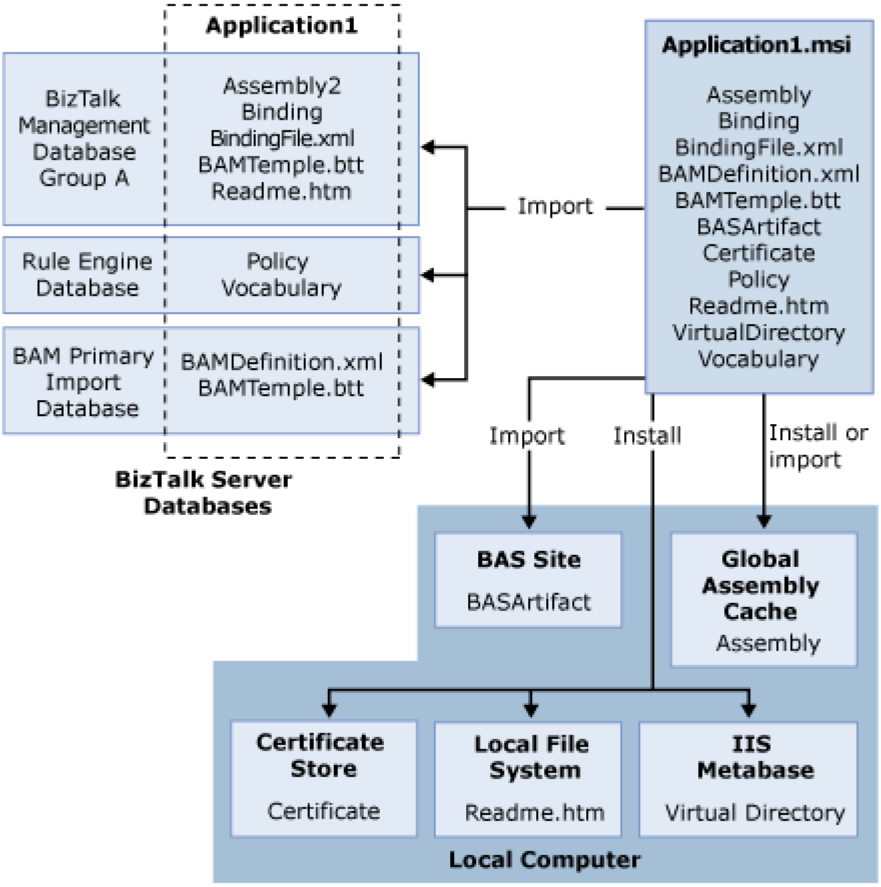
- Scenario: Deploying a New Application
- Scenario: Distributing Artifacts Among Multiple Computers
- Scenario: Updating the Artifacts in a Deployed Application
- Scenario: Deploying Two Versions of an Assembly to Run Side-by-side

Scenario: Deploying a New Application

This topic describes the scenario of deploying an application into a new environment where it hasn't been deployed before; for example deploying an application that has been configured in a staging into a production environment.

As described in The Application Deployment Process, when you want to move an application from one environment to another, you export the application into an .msi file. You then import the .msi file into the BizTalk group in the new environment. You also install the application on the computers in that group that will run the application. You must start the application and also install the application on each computer that will run it before the application can begin functioning.

In following diagram, Application1 is imported from an msi file into BizTalk Group B.



This imports artifacts into the various BizTalk databases as follows:

- The assembly, bindings, binding file, BAM template, BAS artifact, and Readme file are all added to the BizTalk Management database.
- The policy and vocabulary are added to the Rule Engine database.
- The BAM template and BAM definition file are both added to the BAM Primary Import database.

Each of these artifacts is also associated with Application1 in the BizTalk Management database.

The application is also installed on a local computer from the .msi file. This installs various artifacts that are included in the .msi file, as follows

- A virtual directory, named VirtualDirectory, is created in the Internet Information Services (IIS) metabase.
- A certificate is added to the local certificate store.

- A BAS artifact is created in the BAS site.
- A Readme.htm file is created on the local file system.
- A BizTalk assembly is added to the global assembly cache (GAC), as this deployment option was selected for the assembly.

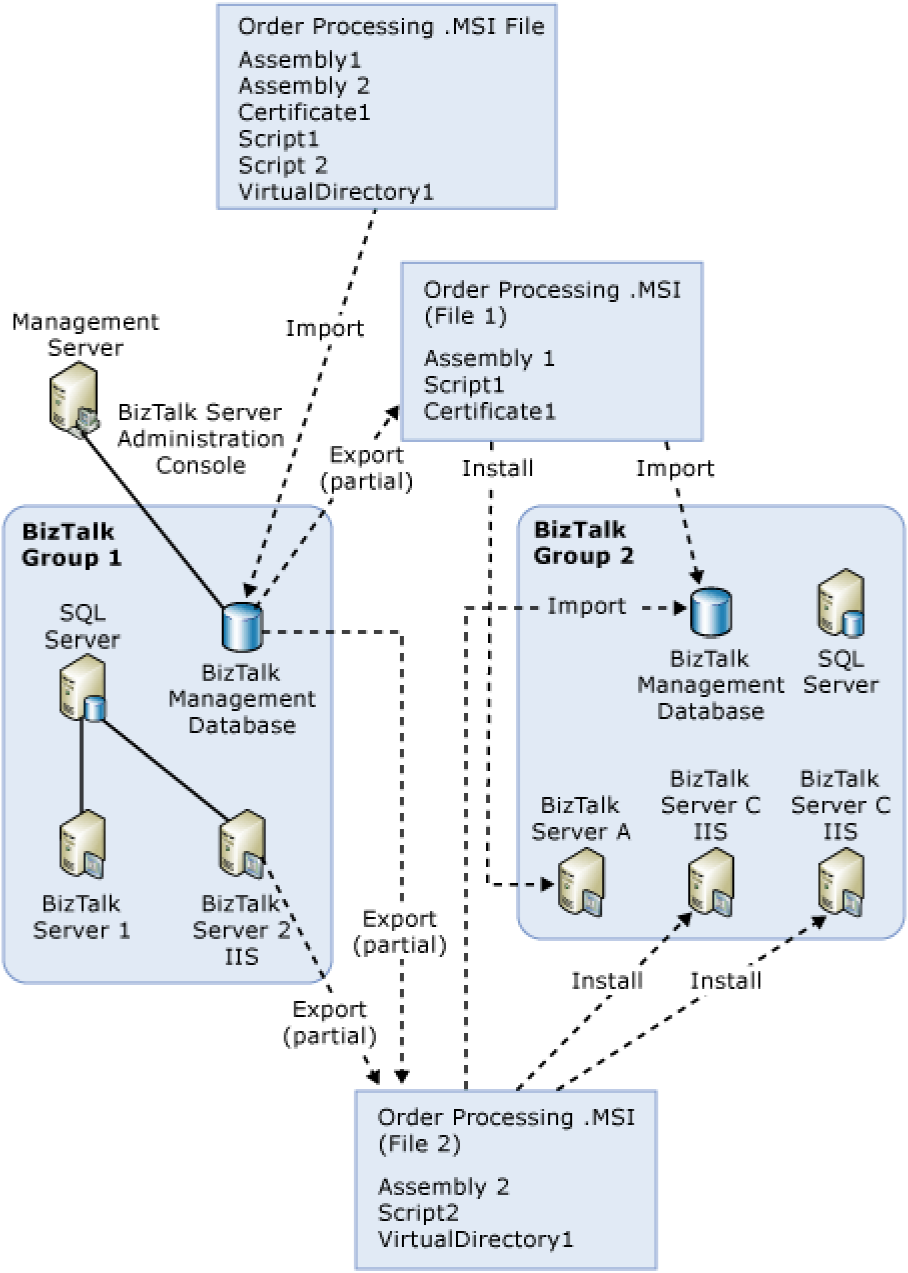
Scenario: Distributing Artifacts Among Multiple Computers

This topic describes the application deployment scenario when the artifacts in an application are selectively installed on different computers. You might want to do this if you want certain assemblies or other types of artifacts in an application to be installed only on specific computers in a BizTalk group. To do this, you can export the artifacts included in an application into multiple .msi files, according to which artifacts you want to install together on a physical computer.

The following diagram shows an .msi file that is imported into the BizTalk Management database for BizTalk Group 1. This creates the Order Processing application and all of its artifacts in that group. The application artifacts are then exported into two different .msi files. One .msi file contains Assembly1, Certificate1, and Script1. The other .msi file contains Assembly2, Script2, and VirtualDirectory1.

Both .msi files are imported into BizTalk Group 2. Because they both belong to the Order Processing application, all of the artifacts in both .msi files are imported into the same application named Order Processing in the new group.

In addition, the application artifacts are installed from the .msi files onto the computers in the group that will run them. Note that the .msi file containing the virtual directory is installed on BizTalk Server B and BizTalk Server C, which are both running IIS.



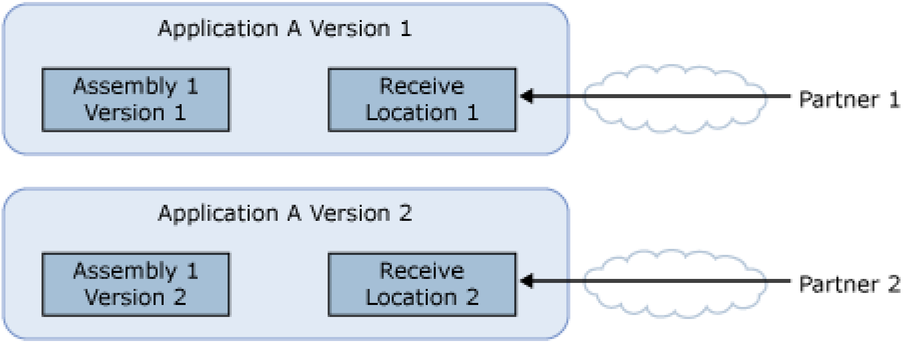
Scenario: Updating the Artifacts in a Deployed Application

This topic describes the scenario of updating the artifacts that are contained in an application that has already been deployed into a production environment.

Scenario: Deploying Two Versions of an Assembly to Run Side-by-side

This topic describes the scenario of deploying two versions of the same assembly within a BizTalk group so that they can both run simultaneously. You can deploy two versions of an assembly side-by-side to maintain two distinct receive locations for processing messages. You may want to do this when you are retiring an assembly until you disable the obsolete assembly. This allows any existing long running transactions to complete successfully. You may also want to do this when you have more than one subscriber for an assembly.

The following diagram shows a typical side-by-side deployment.



For more information about product codes, upgrade codes, and when to change them, see "Patching and Upgrades" in Microsoft Visual Studio 2005 Help.

What Happens to Artifacts During Application Deployment

This section describes what happens to the artifacts associated with an application during the application deployment process. It covers what happens when you add and remove artifacts or export an application, how BizTalk Server imports artifacts into a BizTalk application and the BizTalk Management database, and how artifacts are handled when you install or remove an application on the local computer.

In This Section

- What Happens When Artifacts Are Added and Removed
- What Happens When Artifacts Are Exported
- What Happens When Artifacts are Imported
- What Happens When Artifacts Are Installed and Uninstalled

What Happens When Artifacts Are Added and Removed

This topic describes what happens when you add artifacts to an application. You can add file-based artifacts to an application by using the BizTalk Server Administration console or the BTSTask command-line tool. File-based artifacts include the following types:

- BizTalk assemblies
- .NET assemblies that are not BizTalk-specific
- Binding (.xml) files
- COM components
- Certificates
- Pre- and post-processing scripts
- Policy (.xml) files
- Ad hoc files, such as Readme files
- Virtual directories

When you add an artifact to an application, the artifact is associated with the application in the BizTalk Management database. Any file-based data for the artifact is added to the serialized artifact table, adpl_sat, in the BizTalk Management database. When you export the application into an .msi file, all of the data for the artifact (if you select it for export) is exported from the adpl_sat table into the .msi file. When you import the .msi file, the artifact data is added to the adpl_sat table in the new BizTalk group.

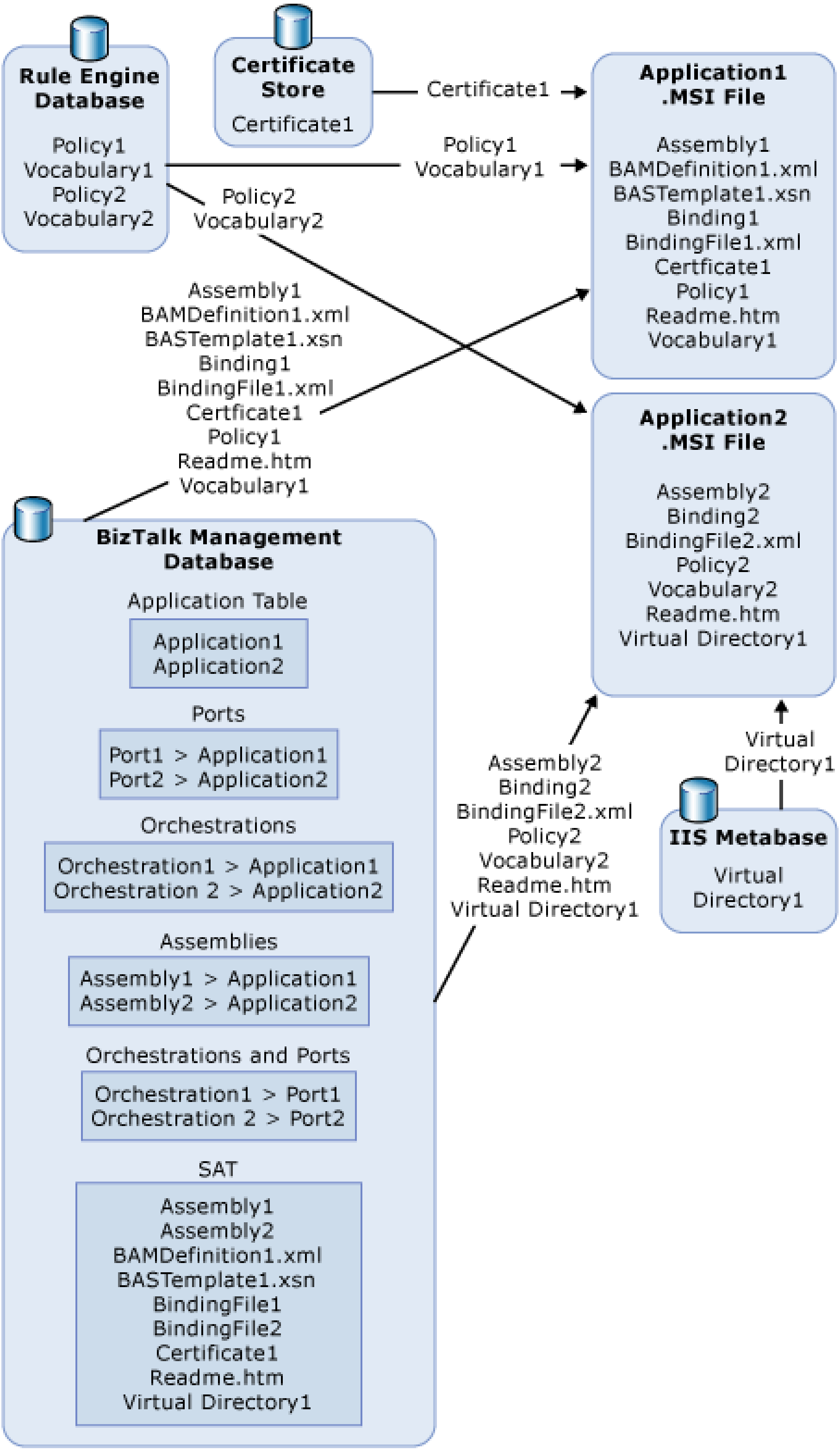
When you add a binding file, you can specify the target deployment environment in which you want the bindings to be applied. Unlike importing a binding file, when you add a binding file, its bindings are not applied immediately. They are applied only when (and if) you export the application containing the binding file into an .msi file and then import the .msi file into another BizTalk group, selecting the deployment environment that you specified for the particular binding file.

BizTalk assemblies are not added to the global assembly cache unless you specify this option when you add the assembly

What Happens When Artifacts Are Exported

You can use the BizTalk Administration console or the BTSTask command-line tool to export a BizTalk application with its artifacts. You can also export a policy or bindings. When you export any of these items, data is exported from the BizTalk Management database and Rule database for the group. In addition, if you export an application that includes an HTTP receive location, the virtual directory associated with it is exported, if you select the virtual directory to export. Also, if you export an application with a receive port that has a certificate associated with it, the certificate is also exported. The artifact files do not need to be present on the local file system to be exported.

The following diagram shows how the artifacts of two different applications that are contained in the BizTalk databases, certificate store, and IIS metabase are exported into two different .msi files.


Rule Engine Database
Policy1
Vocabulary1
Policy2
Vocabulary2
Certificate Store
Certificate1
Certificate1
Policy1
Vocabulary1
Policy2
Vocabulary2
Application1 .MSI File
Assembly1
BAMDefinition1.xml
BASTemplate1.xsn
Binding1
BindingFile1.xml
Certficate1
Policy1
Readme.htm
Vocabulary1
Assembly1
BAMDefinition1.xml
BASTemplate1.xsn
Binding1
BindingFile1.xml
Certficate1
Policy1
Readme.htm
Vocabulary1
Application2 .MSI File
Assembly2
Binding2
BindingFile2.xml
Policy2
Vocabulary2
Readme.htm
Virtual Directory1
BizTalk Management Database
Application Table
Application1
Application2
Ports
Port1 > Application1
Port2 > Application2
Orchestrations
Orchestration1 > Application1
Orchestration 2 > Application2
Assemblies
Assembly1 > Application1
Assembly2 > Application2
Orchestrations and Ports
Orchestration1 > Port1
Orchestration 2 > Port2
SAT
Assembly1
Assembly2
BAMDefinition1.xml
BASTemplate1.xsn
BindingFile1
BindingFile2
Certificate1
Readme.htm
Virtual Directory1
Assembly2
Binding2
BindingFile2.xml
Policy2
Vocabulary2
Readme.htm
Virtual Directory1
Virtual Directory1
IIS Metabase
Virtual Directory1

When you export a BizTalk application, BizTalk Server generates an .msi (Windows Installer) file that contains the application's artifacts. You can specify some or all of the application's artifacts to export. You can use this .msi file to import the application's artifacts, including all of their data, into an application in another BizTalk group. You can also use this .msi file to install the application on a computer.

When you use the administration console to export a policy for either a BizTalk group or application, it generates an .xml policy file containing the policy information. You can import this policy file into another BizTalk group to create the policy there, so that applications in the group can use it. You can also export policy information for an application by using BTSTask. BTSTask, however, does not have a command to export a policy .xml file. Instead, you can export an application .msi file that contains only the policy. You can then import the .msi file into an application in another group.

You can use either the administration console or BTSTask to export bindings for a BizTalk group, application, or assembly. When you do this, BizTalk Server generates an .xml file containing the binding information for the group, application, or assembly. You can then either add this binding file to an application or import it into a BizTalk group or application.

What Happens When Artifacts are Imported

This topic describes what happens to artifacts when they are imported, as follows:

- Importing a BizTalk .msi file containing artifacts into a BizTalk group or application. (For instructions on doing this, see How to Import a BizTalk Application.)

- Importing an .xml policy file into a BizTalk group. (For instructions on doing this, see How to Import a Policy.)

- Importing an .xml bindings file into a BizTalk group or application. (For instructions on doing this, see Importing Bindings.)

Importing a BizTalk .msi file

When you use the BizTalk Server Administration console to import a BizTalk .msi file, the location from which you start the Import MSI Wizard determines whether or not you can create a new application at the same time into which to import the artifacts. If you start the wizard by right-clicking the BizTalk group, you must provide an application name. If an existing application in the BizTalk group has the name you specify, the artifacts in the file are imported into this application; otherwise, a new application having the specified name is created, and the artifacts are imported into it. If you start wizard by right-clicking an application, you cannot specify an application name, and the artifacts are imported into the current application.

When you use the BTSTask command-line tool to import an .msi file, providing an application name is optional. If you do not provide a name, its artifacts are imported into the default application.

When you import an .msi file, the artifacts that it contains are handled as follows:

- References between the artifacts that you import and the specified application (or the default application if none is specified) are created and stored in the BizTalk Management database.

- If during import you added any references from this application to other applications in the group, the reference is added to the BizTalk Management database.

- If you specified this option, existing artifacts in the application are overwritten by artifacts in the .msi file that have the same full name and version number (if applicable).

- Pre- or post-processing scripts that are configured to run on import will run.

- File-based artifact data is serialized and stored in the BizTalk Management database. This includes data for virtual directories, certificates, BAM templates, and BAS artifacts.

- Policy data is stored in the Rule Engine database.

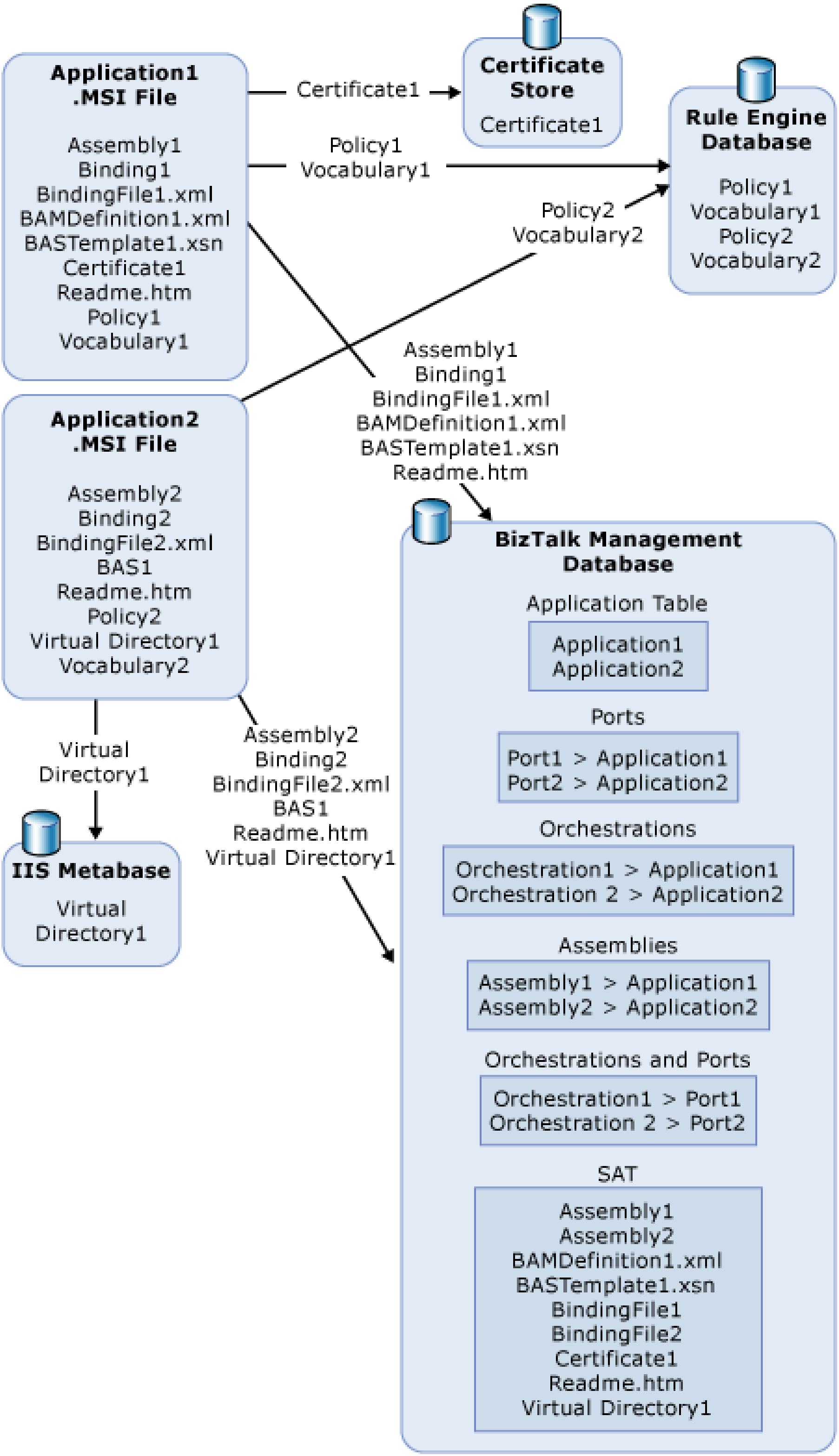
- BAM definition files and templates are stored in the BAM Primary Import database.

- BAS artifacts are added to the BAS site.

- Assemblies that have the "Add to GAC on import" deployment option configured are added to the global assembly cache (GAC).

After you import an application, you can view, manage, and deploy the artifacts in the application either together as a single entity or individually by using the Administration console or BTSTask. For more information, see Application Deployment and Management Tools.

The following diagram illustrates how the artifacts in two different applications are stored in the BizTalk databases, the certificate store, and IIS after the applications are imported.


Application1 .MSI File
Assembly1
Binding1
BindingFile1.xml
BAMDefinition1.xml
BASTemplate1.xsn
Certificate1
Readme.htm
Policy1
Vocabulary1
Certificate1
Certificate Store
Certificate1
Policy1
Vocabulary1
Rule Engine Database
Policy1
Vocabulary1
Policy2
Vocabulary2
Policy2
Vocabulary2
Assembly1
Binding1
BindingFile1.xml
BAMDefinition1.xml
BASTemplate1.xsn
Readme.htm
Application2 .MSI File
Assembly2
Binding2
BindingFile2.xml
BAS1
Readme.htm
Policy2
Virtual Directory1
Vocabulary2
Virtual Directory1
IIS Metabase
Virtual Directory1
Assembly2
Binding2
BindingFile2.xml
BAS1
Readme.htm
Virtual Directory1
BizTalk Management Database
Application Table
Application1
Application2
Ports
Port1 > Application1
Port2 > Application2
Orchestrations
Orchestration1 > Application1
Orchestration 2 > Application2
Assemblies
Assembly1 > Application1
Assembly2 > Application2
Orchestrations and Ports
Orchestration1 > Port1
Orchestration 2 > Port2
SAT
Assembly1
Assembly2
BAMDefinition1.xml
BASTemplate1.xsn
BindingFile1
BindingFile2
Certificate1
Readme.htm
Virtual Directory1

Importing a policy

When you import a policy from an .xml file, the policy is registered in the BizTalk Management database, and its data is added to the Rules Engine database. Unlike importing the policy as part of an application in a BizTalk .msi file, however, the policy is not associated with any application. The policy displays in the Policies node of the <All Artifacts> folder in the BizTalk Server Administration console. After importing the policy you can publish it to make it available for applications in the group to use. For more information, see Managing Policies.

Importing a binding file

When you import a binding file into a BizTalk group, any bindings that currently exist in the group having the same name as bindings in the imported file are overwritten by the bindings in the imported file. The binding file displays in the Resources folder of the <All Artifacts> node in the administration console.

When you import a binding file into a BizTalk application either individually or as part of an application, any bindings that currently exist in the application having the same name as bindings in the file are overwritten by the imported bindings. The binding file displays in the Resources folder of the application into which you imported it.

What Happens When Artifacts Are Installed and Uninstalled

Exporting a BizTalk application creates a Windows Installer (.msi) file that you can use to import the application into a BizTalk group as well as install the application on a computer. To create the application in a BizTalk group, you import the .msi file. To install the application on the local computer, you run the .msi file by double-clicking it. To uninstall an application and its artifacts, you can use the Add or Remove Programs control panel or the UninstallApp command of BTSTask.

The following describes how BizTalk Server installs and uninstalls the different types of artifacts contained in an application:

- **BizTalk assembly.** During installation, the BizTalk assembly file is copied to the destination path, if one was specified when the assembly was added to the application. It also installs the assembly to the global assembly cache (GAC) if this option was specified when the assembly was added. When the application is removed, the assembly file is removed from the file system if it exists there, but is not removed from the GAC, unless the developer included a post-processing script in the application to perform this operation. If necessary, you can remove BizTalk assemblies from the GAC manually. For more information, see How to Uninstall an Assembly from the GAC.

- **NET assembly.** During installation, the .NET assembly file is copied to the destination path, if one was specified when the assembly was added to the application. It also installs the assembly to the GAC and the Windows Registry if these options were specified when the assembly was added. When the application is removed, the assembly file is removed from the file system if it exists there, but is not removed from the GAC or the Windows Registry, unless the developer included a post-processing script in the application to perform this operation. If necessary, you can remove BizTalk assemblies from the GAC or the Windows Registry manually. For more information, see How to Uninstall an Assembly from the GAC and How to Remove Other Files and Settings for a BizTalk Application.

- **File.** During installation, the file is copied to the local file system, if a destination path was specified when the file was added to the application. When the application is uninstalled, the file is removed from the file system.

- **Certificate.** During installation, the certificate is added to the Other People certificate store on the local computer. When the application is removed, the certificate is not removed, unless the developer included a post-processing script in the application to perform this operation. If necessary, you can remove certificates manually. For more information, see How to Remove Other Files and Settings for a BizTalk Application.

- **Virtual directory.** If Internet Information Services (IIS) exists on the computer where the application is being installed, the virtual directory is created; otherwise the virtual directory is not created. If a virtual directory having the same name already exists that uses the specified port, the resources in the application virtual directory are written to the existing virtual directory, and security settings for the existing virtual directory are left unchanged. You should verify that these security settings are sufficient. If the application pool to which the virtual directory is bound does not exist on the local computer when you install the application, the virtual directory is bound to the default application pool.

  When the application is removed, the virtual directory and its files are removed, unless the virtual directory existed before installation or files were added to the virtual directory following installation. In this case, only the files that were installed with the application are removed. The virtual directory and files added to it after application installation are not removed.

- **COM component.** During installation, the COM component is added to the Windows Registry if this option was specified when the component was added to the application. In addition, the file is copied to the local file system, if a destination location was specified when the component was added to the application. When the application is removed, the COM component is not removed from the Windows Registry or the file system unless the developer included a post-processing script in the application to perform this function. If necessary, you can remove COM components from the Windows Registry and file system manually. For more information, see How to Remove Other Files and Settings for a BizTalk Application.

- **Pre-processing and post-processing scripts.** During installation, script files are copied to the local file system, if a destination path was specified when the file was added to the application. Pre-processing scripts run before application import, installation, or removal. Post-processing scripts run after application import, installation, or removal. Script files are removed from the file system when the application is uninstalled. However, if a pre- or post-processing script added files to the file system they are not removed. If necessary, you can remove files manually. For more information, see How to Remove Other Files and Settings for a BizTalk Application.

- **BAM artifacts.** During installation, BAM artifact files are copied to the destination path that was specified when the artifact was added to the application. BAM artifacts are not deployed when the application is installed. When the application is removed, BAM artifact files are removed.

- **BAS artifacts.** BAS artifacts are not installed when an application is installed, and therefore are not affected when you install or uninstall an application that contains them.

Application Deployment Tasks

The topics in this section provide details about the following tasks involved in each phase of the application deployment process for BizTalk Server:

1. **Development.** The developer deploys the BizTalk assemblies that are to be included in the BizTalk application from Visual Studio 2005 into BizTalk Server running on the development computer. This automatically creates the application and populates it with the artifacts contained in the assemblies. Artifacts are all of the items that comprise a BizTalk business solution, including BizTalk assemblies, .NET assemblies, schemas, maps, bindings, certificates, and so forth. The developer completes the application by adding any additional artifacts to it or performing additional configuration. The developer then installs the application from the .msi file, tests it to verify functionality, fixes any issues, and then exports the application from BizTalk Server as an .msi file.

2. **Testing.** The tester imports the .msi file into BizTalk Server running on a test computer, which creates the application in BizTalk Server. The tester also installs the application on the host computers from the .msi file. After testing and bug-fixing is complete, the tester exports the application from BizTalk Server as an .msi file.

3. **Staging.** The IT administrator imports the .msi file into BizTalk Server running on a staging server, configures it for the production environment, installs it on host computers, verifies the functionality, and then exports the finished application as an .msi file.

4. **Production.** The IT administrator imports the .msi file into BizTalk Server running in a production environment, installs the application on the host computers, and then starts the application.

In This Section

- Development Tasks for BizTalk Application Deployment

- Testing Tasks for BizTalk Application Deployment
- Staging Tasks for BizTalk Application Deployment
- Production Tasks for BizTalk Application Deployment

Development Tasks for BizTalk Application Deployment

The following are the steps involved in deploying BizTalk assemblies from Visual Studio 2005 into a BizTalk application, completing the application, and preparing it for deployment to the test environment. This deployment scenario is common in a development environment, where a programmer is developing and debugging a particular BizTalk business solution.

1. **Develop and build the BizTalk assemblies.** You begin by creating your BizTalk business solution in Visual Studio, using orchestrations, schemas, maps, and pipelines. In working on the solution, you build it into one or more BizTalk assemblies. For more information, see Developing BizTalk Server Applications . You also develop and build any non-BizTalk .NET assemblies that are required for your solution to function.

2. **Set deployment properties.** When you are ready to deploy your BizTalk assemblies, you set deployment properties on each Visual Studio project in the solution. In addition to the BizTalk Server properties (Server, Configuration, Database, Redeploy, Restart Host Instances, and Install to Global Assembly Cache), you can also set the Application Name property. This property specifies to which BizTalk application you are deploying each assembly. If Application Name is blank, the assembly is deployed into the default application. For more information, see How to Set Deployment Properties in Visual Studio. You must deploy your non-BizTalk .NET assemblies by adding them to the BizTalk application. This is a separate step, described later, in step 4.

3. **Deploy the BizTalk assemblies to BizTalk Server running on the local computer.** You can deploy the BizTalk assemblies from a menu option, by right-clicking a Visual Studio solution and selecting the Deploy command. This builds the BizTalk assemblies in the projects contained in the solution and adds them to the BizTalk application defined in deployment properties for each project. If the application does not already exist, it is created. The assemblies and their resources, which are called "artifacts," are also deployed into the BizTalk Management database for the group, and you can view and manage them by using the BizTalk Server Administration console or other tools. For more information about this step, see How to Deploy a BizTalk Assembly from Visual Studio.

4. **Add artifacts that are required for the application to function correctly.** From within the BizTalk Server Administration console, you can easily modify an application to complete it, for example by adding and removing artifacts such as send and receive ports, scripts, policies, non-BizTalk .NET assemblies, and so forth. For more information, see Creating and Modifying BizTalk Applications.

5. **Factor the artifacts into multiple applications.** During the development process, you may have deployed your assemblies into a single application for convenience. For various reasons, you may want to factor the artifacts into multiple applications before they are deployed into production. For more information about best practices for factoring applications, see Best Practices for Deploying a BizTalk Application.

6. **Create .msi files for all applications in the solution and install them locally.** You can either use the Export Wizard, available from the BizTalk Server Administration Console or the BTSTask command-line tool to create an .msi file that contains each application's artifacts. For more information, see Exporting BizTalk Applications, Bindings, and Policies. You can take the additional step of installing the artifacts from the .msi files if you want to run the solution on the local computer and verify that it is functioning as expected. For more information, see How to Install a BizTalk Application. Verify that the solution functions as expected.

7. **Redeploy the BizTalk assemblies as needed.** In the process developing and debugging your BizTalk assemblies, you may need to redeploy them multiple times. BizTalk Server 2006 provides a simple mechanism for redeployment. For more information, see How to Redeploy a BizTalk Assembly from Visual Studio.

8. **Export binding files and add them back into the applications (optional).** To make it easier to import the applications back into your development environment at a later time to make changes or additions, you may want to export the bindings for each application and then add them back into the applications, specifying a Development target environment for the bindings. When you later import the application .msi files back into BizTalk Server on your development computer, you can specify that these bindings be applied. For more information, see Binding Files and Application Deployment.

9. **Generate an .msi file for each application to hand off to your test team**. Once you have finished developing and debugging your BizTalk solution, you can use the Export Wizard or BTSTask to generate the application .msi files, as described earlier in step 5. You should import these files into a different BizTalk Server group in your development environment, install them, and then verify that the solution is functioning as expected. You can then hand off to your test team the .msi files, which they can use to import the applications into BizTalk Server running on test computers, as well as to install them, as described in Testing Tasks for BizTalk Application Deployment.

Testing Tasks for BizTalk Application Deployment

The following are the steps involved in deploying the artifacts of a BizTalk application to a test environment for testing and debugging.

1. **Copy the .msi file to your test environment.** As described in Development Tasks for BizTalk Application Deployment, when a BizTalk application is ready to test, the developer exports the application artifacts into an .msi file. A complete

solution may consist of one or more BizTalk applications, so you may receive several .msi files to test. You can copy the .msi file onto your test computer, and then, as described in the next step, import the artifacts from the .msi files into BizTalk Server. You also use the .msi files to install the applications onto the computers that are to run them.

2. **Import the application from the .msi file.** You can use the Import Wizard, available from the BizTalk Server Administration console, to import the artifacts in the .msi file into an application on the current instance of BizTalk Server. For more information, see How to Import a BizTalk Application. If the specified application does not already exist, the importing process creates it. You can then view the application and modify its configuration and contents from the BizTalk Server Administration console. You might need to do this, for example, to modify the bindings for your test environment. For more information, see Creating and Modifying BizTalk Applications. If a binding file was added to the application that has the appropriate bindings for your test environment, you can apply the bindings during the import process. For more information, see Binding Files and Application Deployment.

3. **Install the application.** You can now install the application on servers on which you want to run the application. If the application includes file-based artifacts, you must install it or it will not function. You can install the application by double-clicking the .msi file. This installs and registers the application's artifacts on the local computer so that the application can run. For more information, see How to Install a BizTalk Application.

4. **Test the application.** You can now test the application. After the developer fixes any bugs that you find and gives you an updated .msi file, you can take steps 1, 2, and 3 again.

5. **Export binding files and add them back into the application (optional).** To make it easier to import the application back into your testing environment at a later time if you need to test changes or additions, you may want to export the application bindings and then add them back into the application, specifying a Test target environment for the bindings. When you later import the application .msi file back into BizTalk Server in your test environment, you can specify that these bindings be applied. For more information, see Binding Files and Application Deployment.

6. **Generate an .msi for deploying the application to the staging environment**. Once you have finished testing your BizTalk application, you can deploy it to a staging environment. To do this, you use the Export Wizard to generate a new .msi file, as described earlier in step 3. You can then hand off the .msi file to the IT administrator responsible for staging deployment. The IT administrator will use the .msi file to import the application into BizTalk Server on staging computers and install the application on the computers that will run it, as described in Staging Tasks for BizTalk Application Deployment.

Staging Tasks for BizTalk Application Deployment

The following are the steps involved in deploying a BizTalk application to a staging environment.

1. **Copy the .msi file to your staging environment.** As described in Testing Tasks for BizTalk Application Deployment, when the BizTalk application has completed testing and is ready for staging, the test team provides one or more .msi files for staging. (A complete solution may consist of one or more BizTalk applications, so you may receive several .msi files to stage.) You can copy this .msi file onto your staging computers, and then, as described in the next step, import the BizTalk application from the .msi file into a BizTalk group in the staging environment. You also use the .msi file to install the application onto the computers that are to run the application.

2. **Import the application from the .msi file.** You can use the Import Wizard, available from the BizTalk Server Administration console, to import the contents of the .msi file into an application into a BizTalk group in the staging environment. For more information, see How to Import a BizTalk Application. If the specified application does not already exist, importing the .msi file creates it. You can then view the application and modify its configuration and contents from the BizTalk Server Administration console. You might need to do this, for example, to modify the bindings for your staging environment. For more information, see Creating and Modifying BizTalk Applications. If a binding file was added to the application that has the appropriate bindings for your staging environment, you can apply the bindings during the import process. For more information, see Binding Files and Application Deployment.

3. **Export binding files and add them back into the application (optional).** To make it easier to import the application back into your staging environment at a later time if changes or additions are made to the application, you may want to export the application bindings and then add them back into the application, specifying a Staging target environment for the bindings. When you later import the application .msi file back into BizTalk Server in your staging environment, you can specify that these bindings be applied. For more information, see Binding Files and Application Deployment.

4. **Install the application.** You can now install the application on all of the computers that will run the application. You can install the application by using the Installation Wizard or simply double-clicking the .msi file. For more information, see How to Install a BizTalk Application.

5. **Test the application to verify the configuration and bindings.** You can now test the application to make sure it is functioning as expected in the staging environment. After you or the developer makes any required changes, you can follow steps 1 through 4 again.

6. **Generate an .msi for deploying the application to the production environment**. Once you have finished configuring your BizTalk application for the production environment and verified the configuration, you can deploy it to your production environment. To do this, you use the Export Wizard to generate a new .msi file, as described How to Export a BizTalk Application. You can then hand off the .msi file to the IT administrator responsible for production deployment. The IT administrator will use the .msi file to import the application into BizTalk Server on production computers, as well as install it on the computers that will run it, as described in Production Tasks for BizTalk Application Deployment.

Production Tasks for BizTalk Application Deployment

The following are the steps involved in deploying a BizTalk application to your production environment.

1. **Copy the .msi file to your production environment.** As described in Testing Tasks for BizTalk Application Deployment, when the BizTalk application has completed staging and is ready for production, the IT administrator responsible for staging configuration provides an .msi file to use for production deployment. You can copy this .msi file onto your production computers, and then, as described in the next step, import the BizTalk application from the .msi file into BizTalk Server. You also use the .msi file to install the application onto the computers that are to run the application. A complete BizTalk solution can consist of one or more applications, so you may receive several .msi files to deploy.

2. **Import the application from the .msi file.** You can use the Import Wizard, available from the BizTalk Server Administration console, to import the contents of the .msi file into an application on the current instance of BizTalk Server. For more information, see How to Import a BizTalk Application. If the specified application does not already exist, importing the .msi file creates it. You can then view the application and modify its configuration and contents from the administration console. You might need to do this, for example, to configure certificates and endpoints, such as URLs as well as modify the bindings for your production environment. For more information, see Creating and Modifying BizTalk Applications. If a binding file was added to the application that has the appropriate bindings for the production environment, you can apply the bindings during the import process. For more information, see Binding Files and Application Deployment.

3. **Install and start the application.** You can now install the application on all of the computers that will run the application and start the application in the administration console. You can install the application by double-clicking the .msi file. For more information, see How to Install a BizTalk Application.

4. **Export binding files and add them back into the application (optional).** To make it easier to import the application back into the production environment at a later time to deploy changes or additions, you may want to export the application bindings and then add them back into the application, specifying a Production target environment for the bindings. When you later import the

application .msi file back into BizTalk Server in the production environment, you can specify that these bindings be applied. For more information, see Binding Files and Application Deployment.

5. **Export the application to a new .msi file (optional).** If you made configuration changes or additions to the application, and want the changes to be reflected in the .msi file (for example to deploy the application to other BizTalk groups), you can export a new .msi file, as described in How to Export a BizTalk Application.

Security Considerations for Application Deployment

This section provides security information for BizTalk application deployment.

In This Section

- Security and Windows Installer
- Permissions Required for Deploying and Managing a BizTalk Application
- Application Deployment Security Recommendations

Security and Windows Installer

Windows Installer is a powerful tool for installing and updating BizTalk applications. When using Windows Installer, you should be aware of the security issues that can be created by malicious Windows Installer (.msi) file creators and take steps to prevent them.

Administrators typically run .msi files, and therefore the processes that run during application installation or update have very high privileges. A malicious .msi file creator could exploit this fact in a number of ways, including the following:

- Running script files that make undesirable changes to your system or files.
- Overwriting the COM catalog.
- Overwriting registry values. These changes cannot be rolled back.
- Flooding the file system.

When dealing with .msi files, you should take the following precautions at a minimum:

- Install only .msi files that you completely trust.
- Thoroughly test your .msi files before using them in a production environment.

- Store the .msi files in protected folder that are secured with appropriate discretionary access control lists (DACLs).

Permissions Required for Deploying and Managing a BizTalk Application

Application deployment includes deploying BizTalk assemblies from Visual Studio as well as importing, exporting, and installing BizTalk applications. The basic permissions you need to perform these tasks are as follows:

- As a member of the BizTalk Server Administrators group, you are granted the permissions required to deploy BizTalk assemblies from Visual Studio.

- As a member of the BizTalk Server Administrators group, you are granted the permissions required to import BizTalk applications into a BizTalk group. If the option to add an assembly included in the application to the global assembly cache (GAC) on import has been specified, you must also have Write permissions on the assembly folder. As a member of the local Administrators group, you have this permission.

- As a member of the BizTalk Server Administrators or BizTalk Server Operators group, you are granted the permissions required to:

    - Export BizTalk applications

    - Start and stop send ports, send port groups, and orchestrations

    - Enable and disable receive locations

    - Suspend, resume, and terminate instances

    - Start and stop applications

- As a member of the local Administrators group you are granted permissions to install BizTalk applications on the local computer.

You may want to provide the most restrictive permissions for users to perform these tasks. The remainder of this topic provides more details on the required permissions, as follows.

- Permissions for deploying BizTalk assemblies from Visual Studio

- Permissions for importing an application

- Permissions for exporting an application

- Permissions for installing an application

- Permissions for the Base EDI Adapter

Permissions for deploying BizTalk assemblies from Visual Studio

To deploy BizTalk assemblies from within Visual Studio, you must have Write permission on the BizTalk Management database, at a minimum. You are granted this permission as a member of the BizTalk Server Administrators group.

Permissions for importing an application

To import a BizTalk application, you must have the following permissions, at a minimum. You are granted all of the required permissions as a member of the BizTalk Server Administrator's group, except that if you want to install any assemblies to the GAC, you must also have Write permissions on the assembly folder.

Item	Permissions	When Required
BizTalk Management database	Read/Write	Always required.
BizTalk Rule Engine database	Read/Write	Required only if the application includes rules resources.
BAS database	Read/Write	Required only if the application includes BAS resources.
BAM database	Read/Write	Required only if the application includes BAM resources
Global assembly cache (GAC)	Read/Write	Required only if the application includes assembly resources, and you specify that the assemblies are added to the GAC on import. (See Note.)

Permissions for exporting an application

To export a BizTalk application, you must have the following permissions, at a minimum. You are granted the required permissions as a member of the BizTalk Operators group.

Item	Permissions	When Required
BizTalk Management database	Read	Always required.
BizTalk Rule Engine database	Read	Required only if the application includes rules resources.

BAS database	Read	Required only if the application includes BAS resources.
Certificate store	Read	Required only if the application includes certificate resources.
Internet Information Services	Read	Required only if the application includes virtual directory resources.

Permissions for installing an application

By default, members of the local Administrators group have the permissions required to install BizTalk applications on the local computer. If you want to provide more restricted permissions to users who need to install applications, the following table provides the minimum permissions that you must configure. In addition to these permissions, if your application has resources that require additional permissions to install, such as to create a new database or database table, you must also have these permissions.

Item	Permissions	When Required
Certificate store	Read/Write	Required only if the application includes certificate resources.
Internet Information Services	Read/Write	Required only if the application includes virtual directory resources.
GAC	Read/Write	Required only if the application includes assembly resources, and you specify that the assemblies are added to the GAC on install. (See Note, below.)
File system	Read/Write	Required only if a destination property has been set for a resource.
Registry	Read/Write	Required if the **regsvcs** or **regasm** property is set to True for an assembly resource containing managed COM or COM+ components.
Registry	Read/Write	Required if the application includes unmanaged COM resources

Permissions for the Base EDI Adapter

The EDI subsystem service account must be a member of the BizTalk Host Users group ("BizTalk Application Users" by default) in order for it to be able to access the BizTalk Management database. If the EDI subsystem service account is not able to access the BizTalk Management database, the error message" The EDI Subsystem Service was not started because user [domain\service account user name] does not have sufficient authorization" will appear in the event log and in an error dialog box when you start the EDI subsystem service manually.

Application Deployment Security Recommendations

The following are guidelines for deploying BizTalk applications in your environment:

- All discretionary access control lists (DACLs) are removed from files and folders when an application is exported into an .msi file. After deploying the application from the .msi file, you must reconfigure all security settings on the files and folders, including virtual directories.

- During application deployment, temporary files are stored in the temporary folder for the currently logged on user at <*System Drive*>:\Documents and Settings\<*user*>\My Documents\~tmp. If the files contain sensitive information, you should delete them from the temporary folder following deployment.

- When you export a binding file, BizTalk Server removes the passwords for the bindings from the file. You must reconfigure the passwords before the bindings will function. We also recommend that you mask or remove the password after you use the binding file.

- When you use a binding file that was generated in BizTalk Server 2006, be aware that the trust level of the host is not stored in the binding file. When the bindings are applied during binding import or application import, artifacts are bound to hosts based on the host name only. You should verify that the artifacts have been bound to the correct host and that the trust level is appropriate. For more information, see Binding Files and Application Deployment.

- If the application includes a Web site or an orchestration that uses a Web service, be aware that the security settings on the virtual directory are those in effect when the .msi file is generated during application export. If you are deploying an application to a production environment, then before exporting the application, you should verify that the settings meet your security requirements.

  If, however, you deploy an application that includes a virtual directory, and the virtual directory already exists in the destination environment, the security settings on the existing virtual directory are not changed to match those on the virtual directory you are deploying. You should verify that the security settings on the existing virtual directory meet your requirements.

- If the application pool specified for a Web service does not exist when you import an application, the default application pool is used. The security settings on the default application pool may not be appropriate for your requirements. We therefore recommend that you either create the application pool and configure the appropriate security settings before importing the application, or verify that the settings on the default application pool are appropriate.

- Ensure that you trust the source of the Windows Installer (.msi) files that you deploy to avoid security issues that can be created by malicious .msi file creators. For more information, see Security and Windows Installer.

- Ensure you have the permissions to deploy applications. For more information about the Minimum Security User Rights, see Permissions Required for Deploying and Managing a BizTalk Application.

- Ensure that only BizTalk administrators have access to the assemblies and binding files, as they may contain critical business data such as connectivity and configuration information. If you deploy applications through a network share, configure the discretionary access control list (DACL) on the share containing so that only BizTalk administrators can view its contents.

- When you perform deployment operations, BizTalk Server communicates with the BizTalk Management database. If a firewall exists between them, you must open the appropriate ports on the firewall between the processing, services, and data domains. For more information, see **Required Ports**.

- If you point to a remote location for an assembly or binding file (the latter presenting the larger risk of clear text, sensitive data), you should consider the network between the target file's source computer and the computer you are running deployment from. If the network between these two computers is not fully isolated from potential attackers, you should copy the target file to a removable media and physically transport it to the computer where you run the deployment tool.

- Before you can deploy an application from a network share, you must first change the security configuration of the .NET Framework configuration from local intranet to Full Trust.

Dependencies and Application Deployment

This topic describes how dependencies between artifacts in two or more BizTalk applications affect application deployment and maintenance.

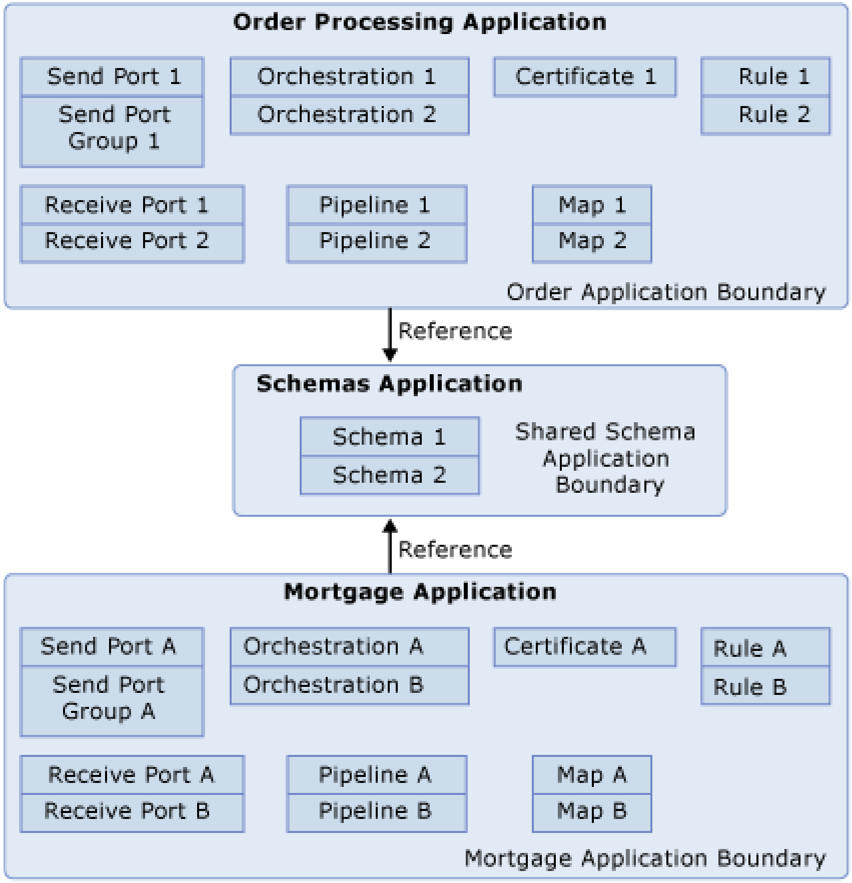
When one artifact needs to use another artifact in order to function properly, it is said to be *dependent* on another artifact. An example of such a dependency is when an orchestration needs to use a specific schema for message resolution or a specific pipeline in order to transmit messages correctly. In both of these scenarios, the orchestration is dependent on another artifact.

Before you can update an artifact in an application, you must first undeploy it, along with any artifacts that depend on it. When artifacts that have dependencies exist in the same application, BizTalk Server automatically handles the undeployment and redeployment tasks for of the updated and dependent artifacts. When artifacts that have dependencies exist in different applications, however, this is not the case. You must take manual steps to undeploy the artifacts that have the dependencies before you can update an artifact on which they depend. Afterwards, you must manually redeploy the dependent artifacts.

To avoid the need to take these manual steps when you want to update an artifact on which other artifacts depend, you can attempt to keep all artifacts with dependencies together in the same application. This is not always possible, however. As described in Artifacts That Must Be Unique in an Application or Group, most types of artifacts must be unique in a BizTalk group. You cannot have the same artifact in two different applications in the same group, even if both applications contain artifacts that depend on the same artifact.

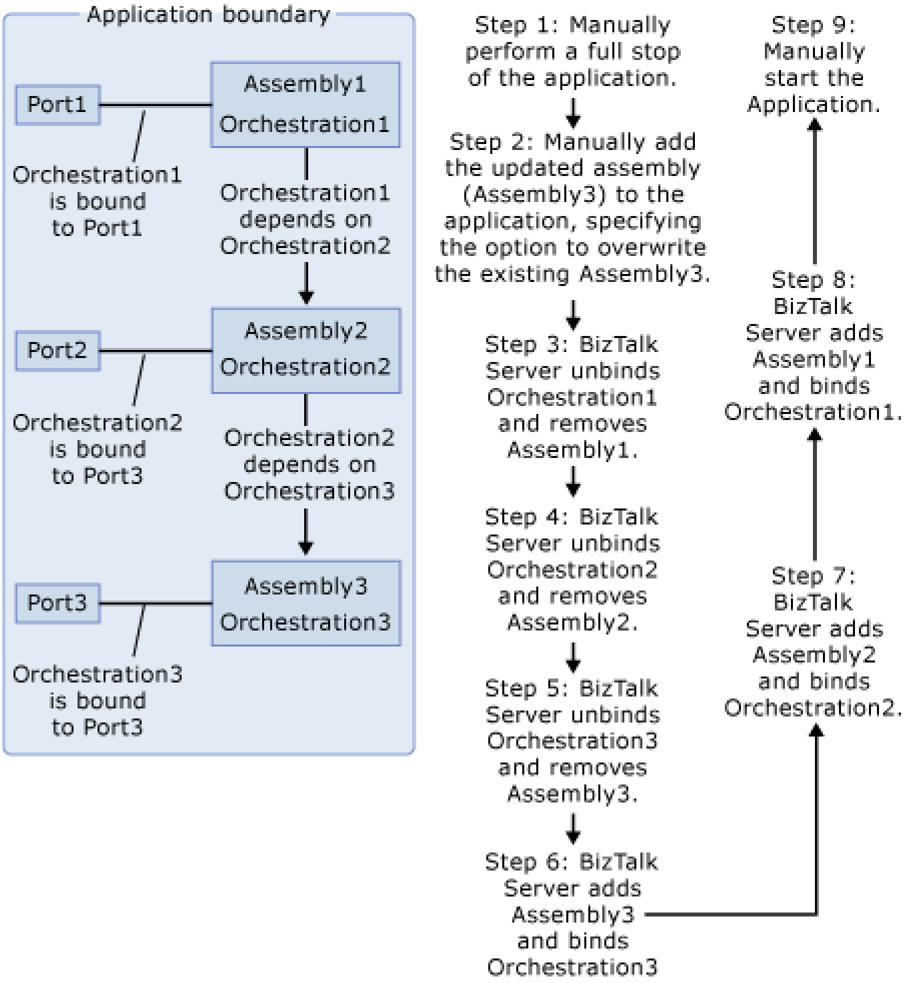
When this happens, you can add the needed artifact to one application and then add a reference to that application from any other applications containing artifacts that depend on it. When you add a reference to an application, artifacts in the application can use any artifacts in the application that it references. For instructions on adding a reference, see How to Add a Reference to Another Application.

The following diagram depicts two applications that each depend on artifacts in a third application. The Order Processing application uses Schema1, which is contained in the Schemas application, so Order Processing application contains a reference to the Schema application. The Mortgage application uses Schema2, which is also contained in the Schemas application, and likewise the former application contains a reference to the latter.

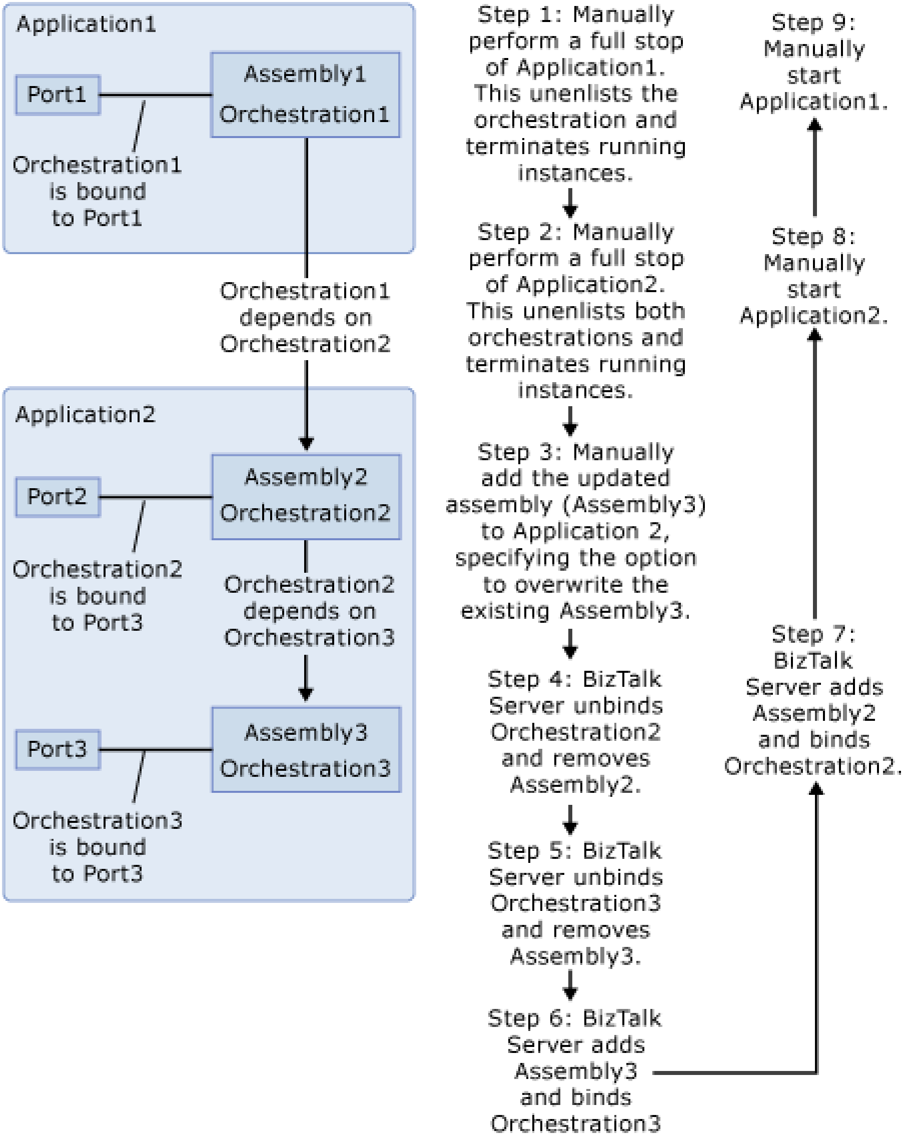


Adding a reference from one application to another creates a dependency between the two applications that affects how you deploy and manage both applications. Because of the various effects of application dependencies described later in this topic, we recommend that you follow the best practices for adding artifacts to application, as described in Best Practices for Deploying a BizTalk Application.

The following diagram illustrates the steps involved in updating an assembly when there is a chain of dependency, and all assemblies that depend on the assembly being updated exist in the same application.


Application boundary
Port1
Assembly1
Orchestration1
Orchestration1 is bound to Port1
Orchestration1 depends on Orchestration2
Port2
Assembly2
Orchestration2
Orchestration2 is bound to Port3
Orchestration2 depends on Orchestration3
Port3
Assembly3
Orchestration3
Orchestration3 is bound to Port3
Step 1: Manually perform a full stop of the application.
Step 2: Manually add the updated assembly (Assembly3) to the application, specifying the option to overwrite the existing Assembly3.
Step 3: BizTalk Server unbinds Orchestration1 and removes Assembly1.
Step 4: BizTalk Server unbinds Orchestration2 and removes Assembly2.
Step 5: BizTalk Server unbinds Orchestration3 and removes Assembly3.
Step 6: BizTalk Server adds Assembly3 and binds Orchestration3
Step 7: BizTalk Server adds Assembly2 and binds Orchestration2.
Step 8: BizTalk Server adds Assembly1 and binds Orchestration1.
Step 9: Manually start the Application.

The following diagram illustrates the steps involved in updating an assembly when there is a chain of dependencies on the assembly being updated, and one of the dependent assemblies exists in a different application.


Application1
Port1
Assembly1
Orchestration1
Orchestration1 is bound to Port1
Orchestration1 depends on Orchestration2
Application2
Port2
Assembly2
Orchestration2
Orchestration2 is bound to Port3
Orchestration2 depends on Orchestration3
Port3
Assembly3
Orchestration3
Orchestration3 is bound to Port3
Step 1: Manually perform a full stop of Application1. This unenlists the orchestration and terminates running instances.
Step 2: Manually perform a full stop of Application2. This unenlists both orchestrations and terminates running instances.
Step 3: Manually add the updated assembly (Assembly3) to Application 2, specifying the option to overwrite the existing Assembly3.
Step 4: BizTalk Server unbinds Orchestration2 and removes Assembly2.
Step 5: BizTalk Server unbinds Orchestration3 and removes Assembly3.
Step 6: BizTalk Server adds Assembly3 and binds Orchestration3
Step 7: BizTalk Server adds Assembly2 and binds Orchestration2.
Step 8: Manually start Application2.
Step 9: Manually start Application1.

The following are effects that may result from dependencies between applications:

- **Stopping an artifact.** If you stop an artifact in one application (which may result from stopping the entire application) that another application depends on, the dependent application will not function correctly. For more information about stopping an application, see How to Start and Stop a BizTalk Application.

- **Removing or changing the state of an artifact.** When you add a reference from one application to another and make any changes to the state of an artifact on which another application depends or remove the artifact, the application that has the dependency will not function correctly. For more information about changing the state of an artifact, see the section on the appropriate artifact in Managing Artifacts.

- **Importing applications that have dependencies.** If you want to import an application into a different BizTalk group and run it in that group, you must also import any artifacts on which this application depends. You can do this by first importing the other application or by adding the needed artifact to the application that requires it. For more information about importing applications, see How to Import a BizTalk Application.

- **Importing applications that have references.** If an application you are importing depends on an artifact in another application, you need to add a reference to this application. The Import Wizard provides this option. If you are using the ImportApp command of BTSTask, however, you must add the reference to the application after import, as described in How to Add a Reference to Another Application. While BizTalk Server verifies that the referenced application exists, we recommend that you take the additional step of verifying that the referenced application contains the required artifact.

- **Installing applications that have dependencies.** When you install an application, you must also install any applications on which it depends. When you install an application that has a dependency on an artifact, such as a BizTalk assembly, which is contained in another application, you must first install the application that contains the artifact. For example, if you want to install Application A, and it depends on an assembly in Application B, you must install Application B first. Then you can install Application A. For more information about installing applications, see How to Install a BizTalk Application.

- **Moving an artifact.** When you move an artifact to a new application, any other artifacts on which it has dependencies are also moved unless the new application has a reference to the application(s) containing the artifacts on which the moved artifact depends. Also, any artifacts that have dependencies on the moved artifact also will be moved unless the application(s) containing them have a reference to the new application. When moving an artifact, you will be shown the list of other artifacts that will also be moved. For instructions on moving an artifact, see How to Move an Artifact to a Different Application.

- **Updating an artifact when an artifact in another application depends on it.** When you update an artifact in an application has a dependency on an artifact in the same application, BizTalk Server automatically takes care of undeploying and redeplying the dependent artifact. If you want to update an artifact in one application, and an artifact in another application has a dependency on it, however, you must undeploy and redeploy the dependent artifact manually as follows:

  1. Stop, unenlist, and unbind the dependent artifact.

  2. Update the artifact on which it depends.

  3. Bind, enlist, and start the dependent artifact.

Binding Files and Application Deployment

This topic provides overview information about using binding files to make BizTalk assembly and application deployment easier. You may find that binding files speed deployment in the following scenarios by avoiding the need to manually configure bindings:

- Moving an application from one deployment environment to another.
- Updating an assembly.
- Deploying an assembly to multiple BizTalk groups.

What is a binding?

A binding creates a mapping between a logical endpoint, such as an orchestration port or a role link, and a physical endpoint, such as a send and receive port or party. This enables communication between different components of a BizTalk business solution. You can create bindings by using the BizTalk Server Administration console.

What is a binding file?

A binding file is an .xml file that contains binding information for each BizTalk orchestration, pipeline, map, or schema in the scope of a BizTalk assembly, application, or group. The binding file describes what host each orchestration is bound to and its trust level as well as the settings for each send port, send port group, receive port, receive location, and party that has been configured. For background information, see Binding Files. You can generate binding files and then apply the bindings they contain to an assembly, application, or group to avoid needing to manually configure bindings in different deployment environments.

Why use binding files?

You may want to use binding files in the following scenarios.

Moving from one environment to another

You can use binding files to make it easier to move an application from one deployment environment to another, such as from a development environment to a test environment. This is because bindings often must be reconfigured for different deployment environments, but by using binding files, you can avoid repeatedly performing this manual configuration step.

One way you can do this is to create a library of bindings from which to select when deploying the application into a new environment. For example, you can create a binding file for your test environment and another one for your production environment and then add them both to the application. When you import the application into the test environment, you can select an option to apply the test bindings. Likewise, when you import the application into the production environment, you can select an option to apply the production bindings. This avoids the need to reconfigure bindings manually for different environments. Another way is to import bindings that you have created for the current environment after importing the application into it. This automatically applies the bindings.

Updating an assembly

When you update an assembly in an application, its bindings are often overwritten or else the assembly may not be bound at all, forcing you to manually reconfigure bindings. To avoid this, you can use a binding file as follows:

- **Updating the same version of an assembly.** If the assembly has early bound ports or dynamic ports, and you changed the port configuration in the BizTalk Server Administration console, the settings will be lost when you update the assembly with an assembly having the same version number. You can export a binding file for the assembly you are going to update. After updating the assembly you can import the assembly into the application and then import its binding file to reapply the previous bindings.

- **Updating an assembly with a newer version.** You can export a binding file for the assembly that you are going to update and then edit it to reflect the new assembly version. After you import the new assembly version into the application, you can import the binding file into the application to apply the bindings. For instructions on editing a binding file, see Customizing Binding Files.

Deploying an assembly to multiple BizTalk groups

When you deploy an assembly into multiple BizTalk groups, you can transport the bindings for the assembly along with the assembly. This avoids the need to separately configure the bindings for the assembly in each group. You can do this as follows:

1. Create a binding file for the assembly that you want to deploy by exporting the assembly's bindings.

2. Add the assembly and its binding file to an application. If you are deploying the assembly separately from other artifacts, the application can contain only the assembly and the binding file.

3. Export an .msi file for the application, being sure to select the binding file to export as well.

4. Import the application .msi file into the BizTalk groups and applications where you want to deploy it. The bindings in the file are automatically applied to the assembly on import.

How can I generate and use binding files?

A binding file is not automatically generated for a BizTalk assembly, application, or group, but you can generate a binding file by exporting bindings, as described in Exporting Bindings. You can then import the binding file into an application or group, as described in How to Import Bindings into a BizTalk Application and How to Import Bindings into a BizTalk Group, which automatically applies its bindings.

Alternatively, you can add the binding file to an application so that its bindings are applied when the application is imported into another group, rather than being applied immediately, as described in How to Add a Binding File to an Application. Using the last method, you can add multiple binding files to an application and optionally specify a target deployment environment for each one. When you import the application, you can then select which bindings to apply, based on the target deployment environment. Using the last method, you can also import separate binding files for the different assemblies in your application.

How are bindings applied?

Bindings are applied when a binding file is imported into an application, or when an application is imported into a new BizTalk group. When using binding files, it is important to understand how artifacts are bound to hosts and in the order in which bindings are applied.

Binding to hosts

When bindings are exported either separately or as part of an application, hosts and trust levels are stored in the binding file as follows:

- **Send Port.** Trust level of the host associated with the send handler.
- **Receive Location.** Trust level of the host associated with the receive handler.
- **Orchestration.** Trust Level of the host.

When the bindings are imported into an application, or an application is imported from the .msi file into a new BizTalk group, hosts and trust levels in the binding files are matched to hosts and trust levels in the application, as follows:

- **Send Port.** The send port is bound to a send handler of the same name and bound to a host with the same trust level as that stored in the binding file.
- **Receive Location.** The receive location is bound to a receive handler of the same name and bound to a host with the same trust level as that stored in the binding file.
- **Orchestrations.** The orchestration is bound to a host with the same name and trust level and as that in the binding file.

Order in which bindings are applied

When you import an application, bindings are applied in the following order:

1. Application bindings generated by BizTalk Server that were not explicitly added to the application via a binding file, but that were explicitly selected by the user for export into the application .msi file.

2. Bindings in binding files that have been added to the application, and which do not have a target deployment environment specified. These bindings are applied in no specific order.

3. Bindings in binding files that have been added to the application, and which have an associated target deployment environment that matches the deployment environment selected for application import. These bindings are applied in no specific order.

As bindings are applied during the import process, bindings that have already been applied are overwritten by new bindings that have the same name. In other words, the last binding of a particular name to be applied takes effect.

For example, if an existing application includes a send port named SendPort1, and a binding file is applied that describes a send port with the same name, the settings in the binding file will overwrite the existing settings for SendPort1. If an existing application includes an orchestration named ErrorHandling.ErrorHandler.ResubmitLogic, for example, and a binding file describes an orchestration having the same name, all of the existing bindings for the orchestration will be written with the bindings in the binding file.

Application Versioning

This topic describes how to update an application with a new version of an assembly, when the assembly contains an orchestration that handles long-running transactions.

In a production environment with running or dehydrated schedules, you can enable outdated assemblies to complete processing and deploy a new version to service new requests. You can remove subscriptions from this undeployed assembly so that no new messages subscribe to it, and send the new messages to the new version. After all messages have been processed by the old assembly, you can undeploy it. Undeployment fails if there are running or dehydrated instances of an orchestration in the assembly.

If you deploy an item (for example, a schema, map, or orchestration in an assembly), and subsequently revise the item and deploy the revised item, your application could still use the outdated item. To prevent this from happening, perform the following steps when you uninstall the assembly that contains the outdated item from the GAC:

- Remove the assembly that contains the outdated item from the GAC

- Deploy the assembly that contains the updated item
- Install the assembly in the GAC

As manual operations can take a few seconds to perform, manually unenlisting an orchestration and then enlisting a new version of it can result in suspended messages. Therefore, we recommend that you perform this operation programmatically. The following code sample illustrates how to use the Explorer OM API to unenlist an existing orchestration and enlist the new version of the same orchestration.

Artifacts That Must Be Unique in an Application or Group

Certain types of artifacts in a BizTalk group or application must be unique, as follows:

- Both BizTalk assemblies and non-BizTalk .NET assemblies must have a unique full name in the group. The full name consists of a file name, public key token, culture, and version.
- Rules and policies must have a unique name and version number in the group.
- Send ports, send port groups, receive ports, and receive locations must have a unique name in the group.
- Web sites must have a unique virtual directory name in the group.
- BAM resources must have a unique file name in the group.
- BAS resources must have unique names in the group, as follows: Any self profile, partner profile, group profile name must be unique in a BizTalk group across all types of profiles (self, partner, group). The agreement name must be unique within a BizTalk group.
- Certificates must have a unique thumbprint in the group.
- COM components must have a unique file name in the group.
- File-based artifacts, such as scripts and other flat files, must have unique file names in the application, but not in the group.

You can specify the overwrite option to import or add an artifact to an application if the artifact already exists in the application and it is of a type that must be unique in an application. If the artifact already exists in another application in the group, and it is a type that must be unique in the group, you can neither add nor import it.

If you need to use an artifact in one application and it already exists in another application in the group, you can create a reference to the application containing the artifact. For more information see How to Add a Reference to Another Application.

Important Considerations for Updating Applications

When you update an application you may cause other applications running in the BizTalk group to stop or malfunction. The following are important issues to consider before updating an application, especially one that is running in a production environment.

- If you add an assembly to an application that includes a new schema with the same message type as an existing schema in the BizTalk group, the schema that was deployed last will be used when schema resolution occurs in pipelines. In addition, if one message type refers to more than one .NET type, this ambiguity may cause pipeline execution failure. This is because schema lookup uses message type, the target namespace, and root name of the instance. This can occur for pipelines in any application that uses a schema with the same message type as the new schema.

- Removing a schema causes the previous version of the schema to become active.

- Parties and rules are visible at a group scope, so adding additional parties and rules could disrupt other applications.

- If you are undeploying an artifact on which another artifact depends, the dependent artifact must be undeployed first.

- If a BizTalk assembly in an existing application is updated, you may need to restart host instances for the changes to take effect. Restarting a host instance stops all other applications that are running on the host instance.

- If you use Visual Studio to deploy an application into a production environment (which we strongly recommend against doing) and the Restart Host Instances option is set to True in project properties, all host instances will restart when you deploy the application. This will stop all other applications that are running on the same host instance.

How to Start and Stop a BizTalk Application

This topic describes how to use the BizTalk Server Administration console to start and stop a BizTalk application.

Before you can start an application, bindings must be configured for all orchestrations it contains, as described in How to Configure Bindings for an Orchestration.

When you start an application, you can select one or more of the following options, as follows:

- Enlist and start all orchestrations.

- Enlist and start all send ports.
- Enlist and start all send port groups.
- Enable all receive locations.
- Start all associated host instances.
- Resume suspended instances.

When you stop an application, you can select one of the following options:

- **Partial Stop - Allow running instances to continue.** This option disables all receive locations in the application, but leaves all other artifacts in their previous state. This allows running instances to complete processing messages that are currently in the system. Note that if a message transaction requires multiple inputs, it may not be able to complete when you use this option.

- **Partial Stop - Suspend running instances.** This option disables all receive locations and stops all orchestrations and send ports in the application. It does not unenlist or undeploy any artifacts. Use this option when you want to temporarily pause the application.

- **Full Stop - Terminate instances.** This option disables all receive locations, stops and unenlists all orchestrations and send ports, and undeploys all policies in the application. Use this option when you want to completely undeploy an application. Note that any running instances that are still processing messages will not complete processing. For more information, see Undeploying BizTalk Applications.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. BizTalk Operators can perform a partial stop, but not a full stop. In addition, BizTalk Operators can start an application is all artifacts are enlisted. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To start or stop a BizTalk application

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 expand Applications.

3. Right-click the BizTalk application that you want to start or stop, and then click **Start** or **Stop**.

4. Select the start or stop options you want and click **Start** or **Stop**.

Creating and Modifying BizTalk Applications

This section describes how to use the BizTalk Server Administration console or the BTSTask command-line tool to create, configure, and modify BizTalk applications.

For background information on creating, managing, deploying, and updating BizTalk applications, see Understanding BizTalk Application Deployment and Management. For information about managing artifacts, including editing their properties, see Managing Artifacts. For instructions on deploying BizTalk applications, see Deploying BizTalk Applications.

In This Section

- How to Create an Application
- How to Configure an Application
- How to Change the Name of an Application
- How to Create or Add an Artifact
- How to Link to a Readme File
- How to Add a 64-Bit Artifact to an Application
- How to Change the Default Application
- How to Add a Reference to Another Application
- How to View the References of an Application
- How to Remove a Reference to Another Application
- How to Move an Artifact to a Different Application
- How to Remove an Artifact from an Application

How to Create an Application

There are four ways to create a new BizTalk application:

- Create the BizTalk application by using the New Application command of the BizTalk Server Administration console or the AddApp command of the BTSTask command-line tool. You can find the procedures for doing this later in this topic.

- Import an .msi file that was exported from another application. If the application does not already exist in the current BizTalk group, the application, including its artifacts, is automatically created in the current BizTalk group. For more information, see How to Import a BizTalk Application.

- Deploy BizTalk assemblies from Visual Studio into a BizTalk application. If the application specified in Visual Studio does not already exist in the current BizTalk group, it is automatically created. For more information, see Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

- Use the BizTalk Explorer Object Model to create a script that creates an application. For more information, see Using the BizTalk Explorer Object Model from Managed Code .

Before creating a new application, you might want to decide how to configure the following options:

- **What to name the new application.** Each application in the BizTalk group must have a unique name.

- **Whether you need to add any references to other applications.** You must add a reference from this application to any applications containing artifacts that you want to reuse in this application. This creates a dependency between the applications, which affects the way that you must deploy them. For background information, see Dependencies and Application Deployment and How to Add a Reference to Another Application.

- **Whether you need to create more than one application.** You may need to deploy some artifacts into separate applications. For example, shared artifacts should be deployed into their own, separate application. For more information, see Best Practices for Deploying a BizTalk Application.

Once you have created an application, you can add artifacts to it and make other modifications, as described in the other topics in this section (Creating and Modifying BizTalk Applications).

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To create a BizTalk application

Using the BizTalk Server Administration console

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**, right-click the BizTalk group in which you want to create the new application, point to **New**, and then click **Application**.

3. In **Name**, type the name of the application. The name must be unique in the BizTalk group.

4. If you want to make this the default application for the BizTalk group, select the **Make this the default application** check box.

5. In **Description**, type a description for the application.

6. If this application will reuse artifacts from another application, click **References** and follow the remaining steps. Otherwise, click **OK** and take no further steps.

7. Click **Add**, select the application that contains the artifacts you want to reuse in this application, and then click **OK.** Repeat this step for any additional applications for which you want to add references.

8. If you want to remove an application from the list, select the application and click **Remove**.

9. When finished, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddApp /ApplicationName:*value* [**/Default**] [**/Description:***value*] [**/Server:***value*] [**/Database:***value*]

   Example:

   BTSTask AddApp /ApplicationName:MyApplication /Description:"My favorite application" /Server:MySQLServer /Database:BizTalkMgmtDb

Parameter	Value
/ApplicationName	Name of the application to be created. Names containing spaces must be enclosed in double quotation marks (").
/Default	When specified, makes the new application the default application for the BizTalk group.
/Description	Description of the application. Descriptions containing spaces

	must be enclosed in double quotation marks (").
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

How to Configure an Application

This topic describes how to use the BizTalk Server Administration console to configure the artifacts in an application, as follows:

- Bind logical ports and role links to physical ports and parties.
- Map orchestrations to the host under which they should execute.
- Configure send and receive ports, send port groups, and receive port locations.

If the application does not contain at least one orchestration, send port, send port group, receive port, or receive location, the information in this topic does not apply.

After completing this configuration, you will be able to start the application, as described in How to Start and Stop a BizTalk Application.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure an application

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, right-click the application, and click **Configure**.

3. In the Orchestrations list in the left pane, click an orchestration, configure properties as described in the following table, and then click **OK.** If there are no orchestrations in this application, the Orchestrations list does not display.

Use this	**To do this**
Host	From the drop-down list, select a host on which to enlist the application.
Bindings - Inbound Logical Ports	Displays the name of a logical inbound port. Logical ports are found on the port surfaces in orchestrations.
Bindings - Receive Ports	From the drop-down list, select the physical receive port that you want to bind to the corresponding inbound logical port. This list includes ports in the current application as well as ports in any referenced applications.
Bindings - Outbound Logical Ports	Displays the name of a logical outbound port. Logical ports are found on the port surfaces in orchestrations.
Bindings - Send Ports/Send Port Groups	From the drop-down list, select the send port or send port group that you want to bind to the corresponding outbound logical port. This list includes ports in the current application as well as ports in any referenced applications.

4. In the Messaging list in the left pane, click **Send Ports and Send Port Groups**, configure properties as described in the following table, and then click **OK**. This list displays all of the send ports and send port groups in the current application.

Use this	**To do this**
New	Click to create a new send port or send port group. Options for send ports are identical to the options available when you right-click the **Send Ports** node in the console tree and point to **New**. For more information, see **How to Create a Send Port**or How to Create a Send Port Group.
Properties	Click to display properties for the selected send port or send port group. If you select a send port, the **Send Port Properties** page displays, where you can configure the port. If you select a send port group, the **Send Port**

	Groups Properties page displays, where you can configure the send port group. For more information on configuring properties, see Managing Send Ports and Send Port Groups.
Name	Displays the name of the physical send ports.
Filter	Displays the filter expression applied to the send port. Filters are set up in the **Send Port Properties** window.
URI	Displays the URI for the selected port.

5. In the Messaging list in the left pane, click **Receive Ports and Locations**, configure properties as described in the following table, and then click **OK.** This list displays all of the receive ports and receive locations in the current application.

Use this	To do this
New	Click to create a new receive port. Options for new receive ports are identical to the options available when you right-click the **Receive Ports** node in the console tree and point to **New**. For more information, see How to Create a Receive Port.
Properties	Click to display properties for the selected item. If you have selected a receive port, the **Receive Port Properties** page displays, where you can configure the port. If you have selected a receive location, the **Receive Locations Properties** page displays, where you can configure the receive location. To select a receive port, click the boldface text to the right of **Receive port**. For more information on configuring properties, see **Managing Receive Ports** and Managing Receive Locations.
Name	Displays the name of the receive location associated with a receive port.
URI	Displays the URI for the selected receive location.

6. In the left pane, click **Role Links**, configure properties described in the following table, and then click **OK.** If there are no role links in the application, the **Role Links** link does not display.

Use this	To do this
Enlist	Click to display the **Enlist Parties** dialog box, where you can select a party to participate in the interchange defined by the role.
Unenlist	Click to remove the selected party from participating in the interchange.
Bind	Click to establish a connection between the role link and the physical ports via a party.

Party	Displays the name of the party or parties that you have enlisted to this role.

How to Change the Name of an Application

This topic describes how to use the BizTalk Server Administration console to change the name of an application. The application name that you use cannot already exist in the group.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To change the name of an application

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, right-click the application whose name you want to change, and click **Properties**.

3. In the **Name** box, type a new name for the application, and then click **OK**.

How to Create or Add an Artifact

After you create a BizTalk application, you can add file-based artifacts (for example BizTalk assemblies, .NET assemblies, scripts, and certificates) from the file system or add policies from the Rule Engine database. You can also create send ports, send port groups, receive locations, and receive ports within the application. Creating or adding artifacts adds them to the BizTalk Management database. You can then deploy the application and its artifacts as a single entity, as described in Deploying BizTalk Applications.

For procedures on adding or creating each artifact type, see the following topics:

- How to Create a Send Port
- How to Create a Send Port Group
- How to Create a Receive Port
- How to Create a Receive Location
- How to Add a Policy to an Application

- How to Add a BizTalk Assembly to an Application
- How to Add a Pre- or Post-processing Script to an Application
- How to Add a File to an Application
- How to Add a Certificate to an Application
- How to Add a COM Component to an Application
- How to Add a .NET Assembly to an Application
- How to Add a BAM Definition File to an Application
- How to Add a Binding File to an Application
- How to Add a Self Profile
- How to Add a Partner Profile
- How to Add a Partner Group
- How to Add an Agreement
- How to Add a Template

How to Link to a Readme File

This topic describes how to use the BizTalk Server Administration console or the command line to add a Readme.htm file that will display when a user clicks a link in Add or Remove Programs control panel.

When you install a BizTalk application, a link to a file named Readme.htm is automatically created in Add or Remove Programs control panel. If a Readme.htm file exists in the application's installation folder, a user can open it by clicking "Click here for support information" link that displays under the name of your application and then clicking "%BTAD_InstallDir%\Readme.htm." (%BTAD_InstallDir% is an environment variable representing the application installation folder.)

So that the link to the Readme.htm file from Add or Remove Programs control panel will function, you should do the following:

- Name the file Readme.htm.
- When you add Readme.htm to your application, specify the application installation folder as the destination location of the file (as described in step 6 below). The default application installation folder is %ProgramFiles%\Generated by BizTalk\ApplicationName. You can express this path as %BTAD_InstallDir%.

If you want to overwrite a Readme.htm file that already exists in the application, specify the Overwrite option. If not specified and a Readme.htm file already exists in the application, the add operation will fail.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a Readme file to an application

Using the BizTalk Server Administration console

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 and the BizTalk group containing the application to which you want to add a Readme.htm file.
3. Expand Applications and the application to which you want to add a Readme.htm file.
4. Right-click the **Resources** folder, point to **Add**, and then click **Resources**.
5. Click **Add**, select the Readme.htm file, and then click **Open**.
6. In **Destination**, the full path of the application installation folder is specified by default, as follows: %BTAD_InstallDir%\Readme.htm. You should not change this path. If the correct path to the application installation folder is not provided, the file is not copied to it during installation, and the link will not work.
7. When finished, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.
2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [**/ApplicationName:***value*] **/Type:System.BizTalk:File** [**/Overwrite**] **/Source:***value* [**/Destination:***value*] [**/Server:***value*] [**/Database:***value*]

   Example:

```
BTSTask AddResource /ApplicationName:MyApplication /Type:System.BizTalk:File
/Overwrite /Source:"C:\Readme Files\MyApplication\Readme.htm"
/Destination:%BTAD_InstallDir%\Readme.htm" /Server:MyServer
/Database:BizTalkMgmtDB
```

Parameter	Value
/ApplicationName	Name of the BizTalk application to which to add the Readme file. If the application name is not specified, the default BizTalk application is used. I the name includes spaces, you must enclose it in double quotation marks (").
/Type	System.BizTalk:File (This value is not case-sensitive.)
/Overwrite	Option to overwrite an existing file. If not specified, and a file already exists in the application that has the same name as the file being added, the AddResource operation fails.
/Source	Full path of the source file, including the file name. If the path includes spaces, you must enclose it in double quotation marks (").
/Destination	%BTAD_InstallDir%\Readme.htm This represents the full path of the application installation folder and Readme.htm file and is the location where the Readme.htm file will be copied during installation. If not provided, the file is not copied to the application installation folder and the Readme.htm link in Add or Remove Programs will not function.
/Server	Name of the SQL Server instance hosting the BizTalk Management database. Required if you specify the Database parameter. If not provided, localhost is used. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.
/Database	Name of the BizTalk Management database. Required if you specify the Server parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.

How to Add a 64-Bit Artifact to an Application

BizTalk Server 2006 includes support for 64-bit applications. This means that you can add 64-bit artifacts to a BizTalk application in the same manner as 32-bit artifacts; however, you may encounter the following issues when installing the following 64-bit artifacts on a 32-bit computer:

- **Virtual directory from a Web server that is running a 64-bit worker process.** The virtual directory will install on a 32-bit computer, but it may not function correctly. A mismatch will be logged.

- **Unmanaged COM or Managed COM+ components marked as 64 bit.** These components will not install on a 32-bit computer.

- **64-bit scripts saved as .exe files.** This type of script may not function correctly.

For general instructions on adding artifacts, see How to Create or Add an Artifact. For instructions on adding specific artifact types, see Managing Artifacts.

How to Change the Default Application

This topic describes how to change the default application by editing the properties of an application in the BizTalk Server Administration console. You can also change the default application by creating a new application and specifying it as the default application, as described in How to Create an Application.

When you install BizTalk Server, a default application named BizTalk Application 1 is created in the BizTalk Management database and displays in the BizTalk Server Administration console. When you upgrade from an earlier version of BizTalk Server, your artifacts are automatically placed in this application. In addition, when you import a Windows Installer (.msi) file by using BTSTask without specifying an application, the artifacts in the .msi file are imported into the default application.

You cannot delete the default application, however, you can change which application is the default. If you change the default application, you can then delete the application that was previously the default, if you want. For instructions, see How to Delete a BizTalk Application from the BizTalk Group.

When you change the default application, all of the artifacts remain in the application that was originally the default. You can explicitly move the artifacts out of the application, if you want. For instructions, see How to Move an Artifact to a Different Application.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To change the default application

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 the application that you want to make the default, and then click **Properties**.

3. Select the **Make this the default application** check box, and click **OK**.

How to Add a Reference to Another Application

This topic describes how to use the BizTalk Server Administration console to add a reference from one application to another application. You can also add a reference to the other application when you import your application by using the Import Wizard, as described in How to Import a BizTalk Application.

You add a reference when you want to use an artifact in the current application that already exists in another application in the same BizTalk group. This is because most types of artifacts must be unique in a BizTalk group. Rather than adding the duplicate artifact to the current application, you add a reference to the other application that already contains the artifact. For more information, see Artifacts That Must Be Unique in an Application or Group.

It is not necessary to add a reference when you want to use a virtual directory or a certificate that already exists in another application. Furthermore, we recommend against doing this because it can create a circular reference.

When you import an application that has dependencies, references can be imported as well. When adding a reference to another application, bear in mind that this affects the way you deploy both applications. For more information, see Dependencies and Application Deployment.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a reference to another application

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, and right-click the application in which you want to create a reference. This is the application in which you want to use an artifact that is contained in another application.

3. Point to **Add** and click **References**.

4. In **Applications**, select the check box of the application to which you want to add a reference (the application containing the artifact or artifacts that you want to use), and then click **OK**.

   The reference is added to the current application. In the console tree, a hand icon is added to the application to which you referred from this application to indicate that it is referenced by one or more other applications.

How to View the References of an Application

This topic describes how to use the BizTalk Server Administration console to view the list of applications to which the current application has references. For more information about adding references, see How to Add a Reference to Another Application. You can also view the list of applications that have references to this application.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view the references for an application

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, right-click the application whose references you want to view, and click **Properties**.

3. In the left pane of the properties page, click **References**.

   The applications to which this application refers are listed in upper section of the right pane. The applications that refer to this application, are listed in the lower section of the right pane.

How to Remove a Reference to Another Application

This topic describes how to use the BizTalk Server Administration console to remove a reference from one application to another application. You remove a reference when you no longer need to use an artifact in the current application that exists in another application in the same BizTalk group. For more information on adding references, see How to Add a Reference to Another Application.

If artifacts in the current application still use artifacts in the referenced application, the operation to remove the reference will fail.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a reference to another application

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, right-click the application from which you want to remove a reference (the one that refers to another application), and click **Properties**.
3. In the left pane of the properties page, click **References**.
4. In the right pane, click the application that you no longer want to reference from this application, click **Remove**, and then click **OK**.

How to Move an Artifact to a Different Application

This topic describes how to move an artifact from one application to another within a BizTalk group by using the Move To Application command of the BizTalk Server Administration console. You might want to do this if you need to use an artifact that already exists in one application in a different application in the same BizTalk group.

When moving an artifact, bear in mind the following important points:

- **The applications must be in the same group.** The Move To Application command only works if the application from which you want to move the artifact is in the same BizTalk group as the application to which you want to move the artifact. If you want to move the artifact to an application in a different group, you can either export the artifact from the application in which it currently exists and then import it into the new application, or add the artifact to the application. For instructions, see How to Export a BizTalk Application, How to Import a BizTalk Application, and How to Create or Add an Artifact. You must then manually remove the artifact from the original application, as described in How to Remove an Artifact from an Application.
- **Moving an artifact may also move artifacts on which it depends or that depend on it.** When you move an artifact to a new application, any other artifacts on which it has dependencies are also moved unless the new application has a reference to the application(s) containing the artifacts on which the moved artifact depends. Also, any artifacts that have dependencies on the moved artifact are

moved unless the application(s) containing them have a reference to the new application. When moving an artifact, you are shown the list of other artifacts that will also be moved.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To move an artifact to a different application

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 expand the application from which you want to move an artifact.

3. Click the folder containing the artifact that you want to move, right-click the artifact, and then click **Move To Application**.

4. In the **Destination application** box, click the arrow, and then click the application to which you want to move the artifact.

   The **Moving Artifacts** box displays the artifact to move, along with all dependent artifacts, which will be moved as well.

5. Click **OK**.

How to Remove an Artifact from an Application

Removing or deleting an artifact deletes it from the BizTalk databases, so that it no longer appears in the administration console or in the list of artifacts for an application generated by the BTSTask ListApp command. It does not remove the artifact from the Windows Registry, the global assembly cache (GAC), a virtual directory, or the file system, if it exists in any of these locations. In the case of send ports, send port groups, receive ports, and receive locations, which exist only in the BizTalk Management database, this operation deletes the artifact entirely. For background information, see What Happens When Artifacts Are Added and Removed

Removing or deleting an artifact can have different effects on the functioning of an application, depending on the artifact type. For more information and instructions on removing or deleting a particular type of artifact, see the appropriate topic, as follows:

- How to Remove an Orchestration from an Application

- How to Delete a Send Port

- How to Delete a Send Port Group
- How to Delete a Receive Port
- How to Delete a Receive Location
- How to Remove a Policy from an Application and the BizTalk Group
- How to Remove a Schema from an Application
- How to Remove a Map from an Application
- How to Remove a BizTalk Assembly from an Application
- How to Remove a Pre- or Post-processing Script from an Application
- How to Remove a .NET Assembly, Certificate, or Other Resource Artifact from an Application
- How to Remove a Self Profile
- How to Remove a Partner Profile
- How to Remove a Partner Group
- How to Remove an Agreement
- How to Remove a Template

Deploying BizTalk Applications

For checklists of tasks that you need to perform in different application deployment scenarios, see BizTalk Application Deployment and Management Checklists. For background information on BizTalk applications and their deployment, see Understanding BizTalk Application Deployment and Management. For information about creating an application and modifying it by adding or removing artifacts or adding a reference to another application, see Creating and Modifying BizTalk Applications. For information about deploying BizTalk assemblies, see Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

In This Section

- Exporting BizTalk Applications, Bindings, and Policies
- Importing BizTalk Applications, Bindings, and Policies
- Using Pre- and Post-processing Scripts to Customize Application Deployment

- How to Install a BizTalk Application
- Monitoring the Progress of Your BizTalk Application Deployment
- Customizing Binding Files

Exporting BizTalk Applications, Bindings, and Policies

This section describes how to export a BizTalk application, bindings, or policies. You can export some or all of the artifacts contained in a BizTalk application, or you can export only its bindings or policies. Exporting an application creates a Windows Installer (.msi) file containing the application artifacts that you selected for export. Exporting bindings or policies creates an .xml file of the bindings or policies. For background information on this process, see What Happens When Artifacts Are Exported.

You can import an application .msi file into another BizTalk group to create a new application that contains the application's artifacts in that group. You can import a binding or policy .xml file into another BizTalk application to use the bindings or policies. How to import artifacts is described in Importing BizTalk Applications, Bindings, and Policies.

In This Section

- How to Export a BizTalk Application
- Exporting Bindings
- How to Export a Policy

How to Export a BizTalk Application

This topic describes how to use the BizTalk Server Administration console or the command line to export an application. Exporting a BizTalk application generates a Windows Installer (.msi) file that contains the application and any of its artifacts that you select to export. The default option is to select all of the application's artifacts, but you can select a subset of them. You can then import the .msi file into another BizTalk group to add the artifacts to an existing application in the new group, update the artifacts in an existing application, or create a new application in the group that contains the artifacts being imported. For more information, see How to Import a BizTalk Application. You also use the .msi file to install the application on the computers that will run it, as described in How to Install a BizTalk Application. If the application includes file-based artifacts, you must also install it before it can begin functioning.

When exporting an application, bear in mind the following important points:

- **Existing bindings are automatically overwritten by imported bindings.** If you do not want the bindings in the application you are exporting to overwrite the bindings in an application into which you are importing an .msi file, then you should not select the binding file as a resource to export. When you import an .msi file that contains a binding file into an existing application, the existing bindings are overwritten by the ones being imported, even when you have not selected the option to overwrite existing artifacts.

- **A user may be making changes to an artifact while you are exporting the application.** If a user is modifying a database-based artifact, such as a BAS artifact, virtual directory, certificate, or policy, while an export operation is in progress, the changes will not be reflected in the exported .msi file. Therefore, we recommend that you schedule export operations during hours when users are not likely to be making changes to these artifacts.

- **The application may have dependencies on another application.** This can affect how you deploy the application. For more information, see Dependencies and Application Deployment.

- **Export will fail if the application contains a policy that has been removed from the Rule Engine database.** When you remove a policy from the Rule Engine database by using the Rule Engine Deployment Wizard, it will display in the administration console in a "Not Published" state, and you will not be able to export the application. For more information about the Rule Engine Deployment Wizard, see How to Deploy and Undeploy Policies and Vocabularies .

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information about permissions, see Permissions Required for Deploying and Managing a BizTalk Application. In addition, the Business Rules Engine must be installed. For more information, see Quick Start Guide to Installing and Configuring BizTalk Server 2006 .

To export an application

Using the BizTalk Server Administration console

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 expand Applications.

3. Right-click the application that you want to export, point to **Export**, and then click **MSI file**.

4. On the Welcome to the Export Wizard page, click **Next**.

5. On the Select Resources page, select the artifacts to export into the .msi file, and then click **Next**.

6. If prompted, on the Specify IIS Hosts page, type the server name of the computer hosting the virtual directory that you want to include, and then click **Next**. You will be prompted to specify the server only if the virtual directory has not been previously added to the BizTalk Management database, such as when it was added to the application or it was imported in an application.

7. On the Dependencies page, review the dependencies for the application, and then click **Next**.

8. On the Destination page, in **Destination application name**, type the application name.

9. In **MSI file to generate**, type the full path for the .msi file, and then click **Export**. Example: C:\MSI\Errorhandling.msi

10. On the Summary page, make a note of the location of the log file for this operation, and then click **Finish**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask ExportApp [**/ApplicationName:***value*] **/Package:***value* [**ResourceSpec:***value* [**/Server:***value*] [**/Database:***value*]

   Example:

   ```
   BTSTask ExportApp /ApplicationName:MyApplication
   /Package:C:/MSI/MyApplication.msi /ResourceSpec:"C:\My
   Files\ResourceSpec.xml" /Server:MySQLServer /Database:BizTalkMgmtDb
   ```

   The artifacts you specified are exported into an .msi file in the location you specified.

Parameter	**Value**
/ApplicationName	Name of the BizTalk application to export. If the application name is not specified, the default BizTalk application is used. If the name includes spaces, it must be enclosed with double quotation marks (").
/Package	Path of the .msi file to be created, including its file name.

/ResourceSpec	Path of the resource specification XML file, including file name. You can specify which artifacts to export by editing the resource specification XML file, which is created when you run the ListApp command with the ResourceSpec parameter, as described in ListApp Command. You must manually edit this file to add the Internet Information Services (IIS) host server name for a virtual directory that you want to export if the Web server is on a remote computer.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

Exporting Bindings

The topics in this section describe how to export bindings for a BizTalk group, assembly, or application into an .xml file. (Bindings define how hosts, send ports, send port groups, receive ports, receive locations, parties are associated with orchestrations, pipelines, maps, and schemas.) You can then import the bindings from the .xml file into another group or application. Importing bindings overwrites any existing bindings of the same name in the group or application. You can also add bindings to an application, which does not overwrite existing bindings. The bindings that you add do not take effect until you import the application.

You may find that using binding files speeds application deployment in the following scenarios by avoiding the need to manually configure bindings:

- Moving an application from one deployment environment to another.
- Updating an assembly.

- Deploying an assembly together with its bindings to multiple BizTalk groups.

For more information about using binding files for these purposes, see Binding Files and Application Deployment.

For more information about importing and adding bindings, see How to Import Bindings into a BizTalk Group, How to Import Bindings into a BizTalk Application, and How to Add a Binding File to an Application.

In This Section

- How to Export Bindings for a BizTalk Group
- How to Export Bindings for a BizTalk Application
- How to Export Bindings for a BizTalk Assembly

How to Export Bindings for a BizTalk Group

This topic describes how to use the BizTalk Server Administration console or the command line to export the bindings for a BizTalk group to an .xml file. You can then import these bindings into a BizTalk group or application, as described in How to Import Bindings into a BizTalk Group and How to Import Bindings into a BizTalk Application. For more information about using binding files, see Binding Files and Application Deployment.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To export bindings for a BizTalk group

Using the BizTalk Server Administration console

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 **Applications**, point to **Export**, and then click **Bindings**.

3. On the Export Bindings page, in **Export to file**, type the absolute path of the .xml file to which to export the bindings.

   Example: C:\Bindings\Group1Bindings_Staging1.xml

4. Ensure that **Export all bindings from the current group** is selected, and then click **OK.**

   The bindings are exported to an .xml file in the location that you specified.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK.**

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask ExportBindings /Destination:*value* **/GroupLevel** [**/GlobalParties**] [**/Server:***value*] [**/Database:***value*]

   Example:

   BTSTask ExportBindings /Destination:"C:\Binding Files\MyBindings.xml" /GroupLevel /Server:MyDatabaseServer /Database:BizTalkMgmtDb

Parameter	**Value**
/Destination	Full path of the binding file to create, including the file name. If a binding file having the same path already exists, it is overwritten. If there are spaces in the path, you must enclose it in double quotation marks (").
/GroupLevel	When specified, all bindings in the current BizTalk group are exported.
/GlobalParties	When specified, exports global party information for the group.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.

/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

How to Export Bindings for a BizTalk Application

This topic describes how to use the BizTalk Server Administration console or the command line export the bindings for a BizTalk application to an .xml file. You can then import the binding file into another application, which overwrites the current bindings in the application with the imported bindings of the same name. For more information, see How to Import Bindings into a BizTalk Application. For more information about using binding files, see Binding Files and Application Deployment.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Server Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To export bindings for a BizTalk application

Using the BizTalk Server Administration console

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 expand Applications**.**

3. Right-click the application whose bindings you want to export, point to **Export**, and then click **Bindings**.

4. On the Export Bindings page, in **Export to file**, type the absolute path of the .xml file to which to export the bindings.

   Example: **C:\Bindings\Application1Bindings_Staging1.xml**

5. Ensure that **Export all bindings from the current application** is selected, and then click **OK**.

6. To export all party information for the group, select the **Export Global Party information** check box.

   The bindings are exported into an .xml file in the location that you specified.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask ExportBindings /Destination:*value* [**/ApplicationName:***value*] [**/GlobalParties**] [**/Server:***value*] [**/Database:***value*]

   Example:

   BTSTask ExportBindings /Destination:"C:\Binding Files\MyBindings.xml" /ApplicationName:MyApplication /Server:MyDatabaseServer /Database:BizTalkMgmtDb

Parameter	**Value**
/Destination	Full path of the binding file to create, including the file name. If a binding file having the same path already exists, it is overwritten. If there are spaces in the path, you must enclose it in double quotation marks (").
/ApplicationName	Name of the application from which to export bindings. The application must exist. To verify the application name, you can use the **ListApps** command, as described in ListApps Command. If this parameter is not specified, the default BizTalk application is used. If there are spaces in the name, you must enclose it in double quotation marks (").
/GlobalParties	When specified, exports global party information for the group.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.

/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

How to Export Bindings for a BizTalk Assembly

This topic describes how to use the BizTalk Server Administration console or the command line to export the bindings for a BizTalk assembly to an .xml file. You can then import these bindings into a BizTalk application, which overwrites existing bindings with the imported bindings of the same name. You might want to export the bindings for an assembly before you update it, so that you can import the bindings after you update it to reapply them. For more information about updating applications and assemblies, see Application Versioning. For more information about using binding files, see Binding Files and Application Deployment.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Server Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To export bindings for a BizTalk assembly

Using the BizTalk Server Administration console

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 expand Applications.

3. Right-click the application containing the assembly, point to **Export**, and then click **Bindings**.

4. On the Export Bindings page, in **Export to file**, type the absolute path of the .xml file to which to export the bindings.

   Example: **C:\Bindings\MyAssemblyBindings_Staging1.xml**

5. On the Export Bindings page, click **Export bindings for the selected assembly**, and then in **Assembly**, click the assembly.

6. If you want to export all of the party information in the group, select **Export Global Party Information**.

   The bindings are exported to an .xml file in the location that you specified.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask ExportBindings /Destination:*value* **/AssemblyName:***value* [**/GlobalParties**] [**/Server:***value*] [**/Database:***value*]

   Example:

   ```
   BTSTask ExportBindings /Destination:"C:\Binding Files\MyBindings.xml"
   /AssemblyName:"ErrorHandling.ErrorHandler, Version=1.0.0.0, Culture=neutral,
   PublicKeyToken=11e921d58826420e" /Server:MyDatabaseServer
   /Database:BizTalkMgmtDb
   ```

Parameter	**Value**
/Destination	Full path of the binding file to create, including the file name. If a binding file having the same path already exists, it is overwritten. If there are spaces in the path, you must enclose it in double quotation marks (").
/AssemblyName	Locally unique identifier (LUID) of the assembly from which to export bindings. You can obtain a list of LUIDs for the artifacts in an application by using the ListApp Command.
/GlobalParties	When specified, exports global pary information for the group.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL

	Server is used.

How to Export a Policy

This topic describes how to use the BizTalk Server Administration console or the command line to export one or more policies and associated vocabularies.

When exporting a policy, bear in mind the following important points:

- Using the BizTalk Server Administration console, you can export policies from a BizTalk group or a BizTalk application as well as the vocabularies to export. Using BTSTask, you can export policies from an application, and all of the associated vocabularies will be exported as well. You cannot select the vocabularies to export.

- You can then import the policy or policies into a different BizTalk group or an application in a different BizTalk group, as described in How to Import a Policy.

- Before you can export a policy, it must exist in the Rule Engine database for the BizTalk group. There are several ways to import a policy into the Rule Engine database, as described in How to Import a Policy.

- When you use the administration console for exporting, the policies and vocabularies are exported into an .xml file. When you use the BTSTask command-line tool for exporting, the policies and vocabularies are exported into an application .msi file.

- BTSTask does not provide a specific command for exporting policies; however you can use the ExportApp command of BTSTask to selectively export only the policies you want, and no other artifacts. This generates an application .msi file containing the policies. You can use the ImportApp command to import the .msi file into a different BizTalk group.

Prerequisites

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

- You must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information about permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

- The Business Rule Engine must be installed. For more information, see Quick Start Guide to Installing and Configuring BizTalk Server 2006 .

- The policy that you want to export must exist in the Rule Engine database for the BizTalk group. If you want to export the policy from an application, it must have also been added to the application as well.

To export a policy

Using the BizTalk Server Administration console

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 and expand the BizTalk group.

3. If you want to select the policies to export from all of the policies in a BizTalk group right-click the **Applications** folder, click **Export**, and then click **Policies**.

   OR

   If you want to export the policies in a particular application, expand the Applications folder, right-click the application, click **Export**, and then click **Policies**.

   OR

   If you want to export only a particular policy, click the Policies folder that contains the policy, right-click the policy, and then click **Export**.

4. On the Export Policies page, in **Policies to export**, select the policies to export.

5. In **Vocabularies to export**, select the check boxes of the vocabularies to export, and clear the checkboxes of any vocabularies you do not want to export. The vocabularies used by this policy are automatically selected.

6. In **File to export** into, type the path of the XML file to which to export the policy or policies, and then click **OK**.

Using the command line

1. Use the BTSTask ListApp command with the /ResourceSpec option to generate an XML file that lists the artifacts in the BizTalk application from which you want to export a policy, as described in ListApp Command.

2. Edit the XML file generated in the previous step, deleting all of the artifacts except for the policy or policies that you want to export.

3. Use the BTSTask ExportApp command, and specify the modified XML file for the /ResourceSpec parameter. For more information, see ExportApp Command.

   BTSTask exports the specified policies and all of their associated vocabularies into an application .msi file.

Importing BizTalk Applications, Bindings, and Policies

The topics in this section describe how to import BizTalk applications, bindings and policies into a BizTalk group or application. As mentioned in How to Export a BizTalk Application, exporting an application creates a Windows Installer (.msi) file that you can then use to import the application's artifacts into an application in a different BizTalk group. If the application that you specify for the import does not already exist in the group, the application is created. In addition, application's artifacts are registered and their data stored in the BizTalk databases of the group. For more information, see What Happens When Artifacts are Imported.

You can also import into a BizTalk group or application the bindings that you exported from a BizTalk group, application, or assembly, as described in Exporting Bindings. When you import bindings, they overwrite any bindings of the same name in the BizTalk group, application, or assembly.

In addition, you can import a policy into a BizTalk group that you exported from a BizTalk group or application, as described in How to Export a Policy. Applications in the new group can then use the policy.

In This Section

- How to Import a BizTalk Application
- Importing Bindings
- How to Import a Policy

How to Import a BizTalk Application

This topic describes how to use the BizTalk Server Administration console or the command line import a BizTalk application into a BizTalk group. Importing a BizTalk application registers the artifacts in the BizTalk Management database and writes the data of the artifacts to the appropriate BizTalk databases. For details, see What Happens When Artifacts are Imported. Importing an application does not install the application. You must install an application that includes file-based artifacts before it can run.

If an application having the same name does not already exist in the group, importing the application also creates the application. If the application does already exist in the group, this operation imports the application's artifacts into the existing application. After importing the artifacts, you can view them in the appropriate folder under the application's folder in the BizTalk Server Administration console. You can also view a list of artifacts in the application by using BTSTask, as described in ListApp Command.

Prerequisites

To import a BizTalk application, you must be logged on with an account that is a member of the BizTalk Server Administrators group. To install a BizTalk application, you must also have Write permissions on the local file system. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

Considerations for importing applications

When importing an application, the following considerations may apply:

- **Existing bindings are always overwritten by imported bindings.** When you import an .msi file that contains bindings into an existing application, the existing bindings are overwritten by imported bindings that have the same name. This is the case even when you have not selected the option to overwrite existing artifacts when importing the .msi file. If you do not want the bindings in the application you are exporting to overwrite the bindings in an application into which you are importing the .msi file, then you should not select the binding file as a resource to export during the export operation. For more information, see How to Export a BizTalk Application.

  As bindings are applied during the import process, bindings that have already been applied are overwritten by new bindings that have the same name. In other words, the last binding of a particular name to be applied takes effect. When you import an application, bindings are applied in the following order:

  1. Application bindings generated by BizTalk Server that were not explicitly added to the application via a binding file, but that were explicitly selected by the user for export into the application .msi file.

  2. Binding files that have been explicitly added, and do not have a target deployment environment specified. Bindings in this set are applied in no specific order.

  3. Bindings that have been explicitly added, and that have an associated target deployment environment that matches the deployment environment selected for application import. Bindings in this set are applied in no specific order.

- **The host must exist in the group.** A host corresponding to the host specified in the application bindings contained in the .msi file must already exist in the BizTalk group or the import operation will fail. In addition, the host trust level must match.

- **You may need to add a reference to another application.** If the application you are importing depends on an artifact in another application, you need to add a reference to this application. The application and required artifact must already exist in the group. The Import Wizard provides this option. If you are using the ImportApp command of BTSTask, however, you must add the reference to the application after import, as described in How to Add a Reference to Another Application. For background information, see Dependencies and Application Deployment. The Import Wizard matches the references to existing applications in the group and gives you the option to add a new reference or change an existing reference. You should take the additional step of verifying that the referenced application contains the required artifact.

- **If BAS artifacts are included in the .msi file, you must add a reference to the default application.** This is because certain basic artifacts that are required for BAS to function are created in the default application, BizTalk Application 1, when BizTalk Server is configured with BAS. These artifacts are:

  - A send port named Bas.TpmWebServicePort
  - A receive port named Sts.Outbox
  - A receive location named Sts.Outbox.Location

  If you have changed the default application, then you need to add a reference to the application that contains these artifacts. If you do not do this, the BAS solution will not function correctly.

- **You may need to update file names and receive locations for virtual directories.** If you plan to install on Internet Information Services (IIS) 5.0 virtual directories for Web services that were created for IIS 6.0, after importing the application, you will need to change any unicode characters in file names to ASCII characters. You will then need to update receive location references accordingly. This is because IIS 6.0 fully supports unicode characters in file names, and IIS 5.0 does not. Because changing Web services file names can create errors, we recommend that you always use ASCII characters in file names when you need to move Web services between IIS 6.0 and IIS 5.0.

To import a BizTalk application

Using the BizTalk Server Administration Console

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 **Applications**, point to **Import**, and then click **MSI file**.

3. On the Welcome to the Import Wizard page, in **MSI file to import**, type the path of the .msi file, and then click **Next**. If necessary, you can browse for the .msi file by clicking the **...** button.

4. On the Application Settings page, in the **Application name** drop-down list, select the application name.

5. In **Available applications to add references to**, select the applications to which to add references, if any, and then click **Next**.

6. If you are importing the .msi file into an existing application and want to overwrite artifacts in the existing application, select **Overwrite resources**.

7. On the Application Target Environment Settings page, in the **Target Staging Environment** drop-down list, select the target environment for this application, and click **Next**. This list contains all environments that have been specified for any binding files that have been added to this application. Select <Default> if you want to apply all bindings in the application except for those that have a target environment specified. If the .msi file does not contain a binding file that you want to apply explicitly, you can leave <Default> selected.

8. On the Import Summary page, confirm that the summary information is correct, and then click **Import**.

9. On the Import Succeeded page, if you want to install the application on the local computer, select the **Run the Application Installation Wizard to install the application on the local computer** check box.

10. Click **Finish**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

```
BTSTask ImportApp /Package:value [/Environment:value]
[/ApplicationName:value] [/Overwrite] [/Server:value] [/Database:value]
```

Example:

```
BTSTask ImportApp /Package:"C:\MSI Files\MyApplication.msi"
/Environment:Test /ApplicationName:MyApplication /Overwrite
```

Parameter	Value
/Package	Full path of the .msi file. If the path includes spaces, you must enclose it in quotation marks (").
/Environment	The target deployment environment of the binding file to apply, such as Test. This is the value that was specified for the target deployment environment when the binding file was added to the application.

/ApplicationName	Name of the BizTalk application to which the artifacts in the .msi file are imported. If not specified, the application name that was specified when exporting the .msi file is used. If the specified application does not exist, it will be created. Application names that include spaces must be enclosed with double quotation marks (").
/Overwrite	Option to overwrite artifacts in the application with artifacts in the .msi file that have the same locally unique identifier (LUID). If this option is not specified, and there are one or more artifacts in the application that have the same LUID as artifacts in the .msi file, the import fails. You can view the LUIDs of the artifacts in an application by using the ListApp Command.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

Importing Bindings

The topics in this section describe how to import bindings into a BizTalk group or application.

When importing bindings, bear in mind the following important points:

- **Existing bindings will be overwritten.** If the bindings that you import have the same name as existing bindings, the existing bindings are overwritten. In addition, if the binding file contains parties and aliases, any bindings for parties and aliases of the same name that already exist in the application are overwritten.

- **All bindings in a group must be unique.** If a binding having the same name as one you are importing already exists in the group, the import operation will fail.

- **Binding import behavior depends on the source of the bindings.** The bindings may have been exported from an assembly, application, or group. Depending on from where the bindings were exported, the import operation may have different effects, as shown in the following table.

- **The host must exist in the group.** A host corresponding to the host specified in the bindings must already exist in the BizTalk group into which the bindings are imported or the import operation will fail. In addition, the host trust level must match.

Origin of the bindings	**Importing into an application**	**Importing into a group**
Bindings exported from an assembly	The specified application in must contain an assembly having the same name as the assembly from which the binding file was exported. Otherwise, the import operation fails.	The group must contain an assembly and an application having the same name as the assembly and application from which the binding file was exported. Otherwise, the import fails.
Bindings exported from an application	The application from which the binding file was exported must have the same name as the specified application. Otherwise, the import operation fails.	The application from which the binding file was exported must have the same name as an application in the group into which you are importing the bindings. Otherwise, the import operation fails.
Bindings exported from a group	The group from which the binding file was exported must include an application that has the same name as the specified application. Otherwise, the import operation fails.	Bindings are imported for corresponding applications. In other words, the bindings of an application in the group from which bindings were exported are imported into an application having the same name in the current group.

- **The Name attribute for an adapter may be incorrect.** If the binding file includes settings for an adapter, verify that the Name attribute of the TransportType element in the binding file is the same as that configured for the adapter in the BizTalk Server Administration console (under Platform Settings > Adapters).

  In particular, you should verify that this is the case when importing bindings from BizTalk Server 2006 to BizTalk Server 2006. Some transports for which this may be an issue are as follows:

  - MQS

- MSMQ
- MSMQT
- POP3
- Windows SharePoint Services

In This Section

- How to Import Bindings into a BizTalk Group
- How to Import Bindings into a BizTalk Application

How to Import Bindings into a BizTalk Group

This topic describes how to use the BizTalk Server Administration console or the command line to import bindings into a BizTalk group from an .xml file. For instructions on exporting the bindings from a BizTalk group into an .xml file that you can import, see How to Export Bindings for a BizTalk Group.

You can also import bindings into a BizTalk application, as described in How to Import Bindings into a BizTalk Application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To import bindings into a BizTalk group

Using the BizTalk Server Administration console

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 right-click **Applications**.
3. Point to **Import**, and then click **Bindings**.
4. Click the binding file and click **Open**.

   The artifacts in the binding file are written to the group. They display in appropriate folders of the <All Artifacts> node.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   ```
   BTSTask ImportBindings /Source:value /GroupLevel [/Server:value] [/Database:value]
   ```

   For example, the following command imports bindings into the group defined in the BizTalk Management database running on a SQL Server instance named MyServer.

   ```
   BTSTask ImportBindings /GroupLevel /Server:MyServer /Database:BiztalkMgmtDb /Source:C:\Bindings\Binding1.xml
   ```

Parameter	**Value**
/Source	Full path of the binding file to import, including the file name. Paths that include spaces must be enclosed in quotation marks (").
/GroupLevel	Option to import the binding file into the current group.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

How to Import Bindings into a BizTalk Application

This topic describes how to use the BizTalk Server Administration console or the command line to import bindings into a BizTalk application from an .xml file. You can also import bindings into a BizTalk group, as described in How to Import Bindings into a BizTalk Group.

You can import bindings that have been exported from an assembly, application, or group. If you import group bindings, only the bindings for an application having the same name are applied to the application into which you import the bindings. You can also import bindings from an application that has a different name. For instructions on exporting bindings, see Exporting Bindings.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To import bindings into a BizTalk application

Using the BizTalk Server Administration console

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 expand Applications.

3. Right-click the application into which you want to import bindings, point to **Import**, and then click **Bindings**.

4. Click the binding file and click **Open**.

   The artifacts in the binding file are written to the application. They display in appropriate folders of the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask ImportBindings /Source:*value* [**/ApplicationName:***value*] [**/Server:***value*] [**/Database:***value*]

For example, the following command imports bindings into the application named MyApplication in the default BizTalk group.

```
BTSTask ImportBindings /ApplicationName:MyApplication
/Source:C:\Bindings\Binding1.xml
```

Parameter	Value
/Source	Full path of the binding file to import, including the file name. Paths that include spaces must be enclosed in quotation marks (").
/ApplicationName	Name of the BizTalk application into which the bindings are to be imported. The application must exist or the import operation will fail. If this parameter is not specified, the default BizTalk application is used. Application names that include spaces must be enclosed in quotation marks (").
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

How to Import a Policy

This topic describes how to use the BizTalk Server Administration console to import a policy into a BizTalk group or the BTSTask command-line tool to import a policy into a BizTalk application.

You can create a policy by using the Business Rule Composer, as described in Creating Business Rules , and then import it directly, or you can export a policy from another BizTalk group, as described in How to Export a Policy and then import it.

Importing a policy registers it in the Rule Engine database for the BizTalk group. After you import the policy, you can view it in the BizTalk Server Administration console. If you use the BizTalk Server Administration console to import a policy, it will display in the <All Artifacts> node for the BizTalk group. You can then publish it to make it available to add it to a BizTalk application, as described in How to Publish a Policy. If you use the BTSTask command-line tool to import a policy, the policy will be automatically published and will display in the Policies folder of the application into which you imported it.

When importing a policy, bear in mind the following important points:

- Even if you specify the option to overwrite an existing policy with the imported policy, you cannot import a policy that already exists in the Rule Engine database for the group and has been deployed. The import operation will fail.

- Even if the policy was in a deployed state when exported from another BizTalk group, it will be in an undeployed state when imported.

- BTSTask does not provide a specific command for importing policies; however you can use the ExportApp command of BTSTask to selectively export only the policies in an application that you want, including no other application artifacts. Then you can use the ImportApp command to import the .msi file into an application in a different BizTalk group. This is the approach described in this topic. When you do this, the policy is automatically imported and published in the BizTalk group and added to the specified application.

For more information about working with policies, see Managing Policies. For best practices on adding policies to applications, see Best Practices for Deploying a BizTalk Application.

Prerequisites

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

- You must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

- The Business Rule Engine must be installed. For more information, see Quick Start Guide to Installing and Configuring BizTalk Server 2006 .

- If you want to use the BizTalk Server Administration console to import a policy, you must have available an .xml file containing the policy that you want to import. You can generate such an .xml file by exporting a policy from another BizTalk group or application, as described in How to Export a Policy, or by using the Business Rule Composer, as described in How to Import and Export Policies and Vocabularies .

- If you want to use BTSTask to import a policy, you must have an .msi file containing the policy to import. For instructions, see How to Export a Policy.

To import a policy

Using the BizTalk Server Administration console

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 into which you want to import the policy, and then expand <All Artifacts>.

3. Right-click **Policies**, and then click **Import**.

4. Browse to the .xml file containing the policy and click **Open**.

The policy is imported into the group and displays in the Policies folder of <All Artifacts>.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

BTSTask ImportApp /Package:*value* [**/ApplicationName:***value*] [**/Overwrite**] [**/Server:***value*] [**/Database:***value*]

Example:

BTSTask ImportApp /Package:"C:\MSI Files\MyApplication.msi" /Environment:Test /ApplicationName:MyApplication /Overwrite

Parameter	**Value**
/Package	Full path of the .msi file containing the policy to import. If the path includes spaces, you must enclose it in quotation marks (").
/ApplicationName	Name of the BizTalk application into which to import the policy. If not specified, the application name that was specified when exporting the .msi file is used. If the specified application does not exist, it will be created. Application names that include spaces must be enclosed with double quotation marks (").
/Overwrite	Option to overwrite policies in the application with artifacts in the .msi file that have the same name and version number. If this option is not specified, and there are one or more policies in the application that have the same name and version number as policies in the .msi file, the import fails. You can view the name

	and version number of the policies in an application by using the ListApp Command.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

Using Pre- and Post-processing Scripts to Customize Application Deployment

The topics in this section describe how to create pre- or post-processing scripts to perform custom actions when an application is imported, installed, or uninstalled. Pre-processing scripts perform an action or set of actions before application import, installation, or uninstallation begins. Post-processing scripts perform an action or set of actions after application import, installation, or uninstallation completes.

You might, for example, want to include a pre-processing script that will run before installation to back up resource files before they are overwritten during installation. Or you might want to run a post-processing script to add a certificate to the certificate store after an application is installed.

In This Section

- Creating a Pre- or Post-processing Script
- Valid Combinations of Environment Variables
- What Happens to File Artifacts During Deployment
- Pre- and Post-processing Script Environment Variables

Creating a Pre- or Post-processing Script

You can create a script to perform custom actions when an application is deployed, and then define when it will run during the deployment process. You can include both installation and cleanup code in the same script, using environment variables to delimit the code. You can also pass command-line arguments into the script.

Specifying when a script will run during deployment

You specify when a script will run when you add the script to an application by adding it as either a System.BizTalk:PreProcessingScript (pre-processing script) or a System.BizTalk:PostProcessingScript (post-processing script).

Pre- and post-processing scripts run as follows:

- Pre-processing scripts run at the beginning of the import or installation process.
- Post-processing scripts run at the end of the import or installation process.
- During uninstallation, all scripts run in the opposite order that they run during application installation. Therefore, post-processing scripts run at the beginning of uninstallation and pre-processing scripts at the end of uninstallation.
- If an installation or import fails, scripts are called in reverse order with the appropriate rollback action.

For instructions on adding a script to an application, see How to Add a Pre- or Post-processing Script to an Application.

Once invoked, a pre- or post-processing script determines which deployment state (install, import, delete, or uninstall) it is running in by checking the environment variables BTAD_ChangeRequestAction, BTAD_InstallMode, and BTAD_HostClass. For more information about the variables, see Pre- and Post-processing Script Environment Variables and **Valid Combinations of Environment Variables**

Supported script file extensions

The following script file extensions are supported: .com, .exe, .bat, .cmd, .vbs, .vbe, .js, .jse, .wsf, and .wsh. This set of extensions is defined in the PATHEXT environment variable.

Including installation and cleanup code in the same script

Because scripts run in the opposite order during installation and uninstallation, you can include installation and cleanup code in the same script, qualified with appropriate flag. For example, you can qualify installation code with the following code:

You can qualify cleanup code with the following code:

Passing in command-line arguments

You can pass command-line arguments into the script by specifying the following parameter when you use the BTSTask AddResource command to add the script to the application. When you do this, the arguments are passed to the script when the script is invoked.

/Property:Args="*argument list*"

For instructions on using the AddResource command to add a script to an application, see AddResource Command: Preprocessing Script and AddResource Command: Postprocessing Script

Valid Combinations of Environment Variables

Once invoked, a pre- or post-processing script can determine in which state (install, import, delete, or uninstall) it is running by checking the environment variables BTAD_ChangeRequestAction, BTAD_InstallMode and BTAD_HostClass.

The following table describes the valid combinations of the three variables.

Deployment Action	**Expected Values**		
	BTAD_ChangeRequestAction	BTAD_InstallMode	BTAD_HostClass
Import without overwrite flag	Create	Import	ConfigDb
Import with overwrite flag	Update	Import	ConfigDb
Update (always with overwrite flag)	Update	Install	Host Instance
Uninstall	Delete	Uninstall	Host Instance

What Happens to File Artifacts During Deployment

The following table summarizes the status of the artifact files at each point in the deployment process.

Point in the Deployment Process	Status of Application Artifact Files
Pre-installation	Only files of the type System.BizTalk.File exist on the local file system.*
Post-installation	All files exist on the local file system.*
Pre-uninstallation	All files exist on the local file system.*
Post-uninstallation	All files have been deleted from the local file system.
Pre-import	No files exist on the local file system unless the application has been installed on the local computer.
Post-import	No files exist on the local file system unless the application has been installed on the local computer.

* Files exist on the local file system only if a valid destination location was specified when the file was added to the application.

Pre- and Post-processing Script Environment Variables

The topics in this section describe the environment variables that pre- and post-processing scripts can access.

In This Section

- BTAD_ChangeRequestAction
- BTAD_HostClass
- BTAD_InstallMode
- BTAD_InstallDir
- BTAD_ApplicationName
- BTAD_SilentMode
- BTAD_Server
- BTAD_Database

BTAD_ChangeRequestAction

BTAD_ChangeRequestAction contains a value indicating a change request action.

Remarks

The following table describes the possible values for BTAD_ChangeRequestAction.

Value	Description
Create	Import without overwrite specified
Update	Import or install with overwrite specified
Delete	Uninstall

BTAD_HostClass

BTAD_HostClass indicates whether the operation is being performed on the BizTalk Management database or the local computer.

Remarks

The following table describes the possible values for BTAD_HostClass.

Value	Description
ConfigurationDb	Specifies that the import operation is being performed on the BizTalk Management database for the group
BizTalkHostInstance	Specifies that the install or uninstall operation is being performed on the local computer

BTAD_InstallMode

BTAD_InstallMode describes the installation mode for BizTalk application deployment.

Remarks

The following table describes the possible values for BTAD_InstallMode.

Value	Description
Install	Create or update to BizTalkHostInstance (the local computer)
Import	Create or update to ConfigurationDb (the BizTalk Management database for the group)

Uninstall	Delete from BizTalkHostInstance (the local computer)

BTAD_InstallDir

BTAD_InstallDir contains the installation path of the BizTalk application. This is only available when the action is Create or Update, and BTAD_HostClass is BizTalkHostInstance.

Remarks

BTAD_InstallDir is set to the TARGETDIR property of the Windows Installer package. The default value is %ProgramFiles%\Generated by BizTalk\<*application name*>.

BTAD_ApplicationName

BTAD_ApplicationName contains the name of the BizTalk application

BTAD_SilentMode

BTAD_SilentMode specifies options for running the script in silent mode.

Remarks

When you specify an option for silent mode, BTS Installer places its value into the BTAD_SilentMode variable.

The default value of BTAD_SilentMode is 2, which specifies that the script runs in silent mode. The following table describes the possible values for BTAD_SilentMode.

Value	Description
0	Does not change the user interface (UI) level.
1	Uses the default UI level.
2	Performs a silent installation (the default).
3	Provides simple progress and error handling.
4	Provides authored UI; suppresses wizards.
0x80	If combined with any above value, Windows Installer displays a modal dialog box if the script has executed successfully or if there has been an error. No dialog box is displayed if the user cancels the operation.
0x40	If combined with the 3 value, Windows Installer displays progress dialog boxes

	but does not display any modal dialog boxes or error dialog boxes.
0x20	If combined with the 3 value, Windows Installer displays progress dialog boxes but does not display any modal dialog boxes or error dialog boxes.
0x100	If combined with the 3 value, Windows Installer displays progress dialog boxes but does not display any modal dialog boxes or error dialog boxes

BTAD_Server

When pre- and post-processing scripts run on application import, the BTAD_Server environment variable is set. It specifies the name of the SQL Server instance running the BizTalk Management database for the group.

BTAD_Database

When pre- and post-processing scripts run on application import, the BTAD_Database environment variable is set. It specifies the name of the BizTalk Management database for the group.

How to Install a BizTalk Application

This topic describes how to install an application on the local computer by double-clicking the Windows Installer (.msi) file for the application in the Windows interface or by running msiexe from the command line. You can also start the Installation Wizard as the last step of the Import Wizard, as described in How to Import a BizTalk Application.

Before you can put an application into operation, you must install it on the BizTalk Server computers that will run it. Installing an application places its resources on the local file system. Depending on the application, its contents, and its configuration, installation may also do the following:

- Add assemblies to the global assembly cache (GAC)
- Install certificates and virtual directories
- Add components to the Windows registry.
- Run pre- or post-installation scripts, if they are present in the .msi file.

For background information, see What Happens When Artifacts Are Installed and Uninstalled.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that has Write permissions on the local file system. Depending on the items included in the application, you may also need Write permissions on the Windows Registry, GAC, the certificate store, and Internet Information Services. The Administrators account on the local computer has these permissions. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

Considerations for installing an application

When installing an application, the following considerations may apply:

- **You must also install any applications on which this application has a dependency.** When you install an application that has a dependency on an artifact, such as a BizTalk assembly, which is contained in another application, you must also install the application that contains the artifact. You must do this before you can run the application. For example, if Application A depends on an assembly in Application B, you must also install Application B. Then you can install Application A. For background information, see Dependencies and Application Deployment.

- **You should stop an application that you are updating.** If you are performing the installation to update an artifact in the application, you do not need to stop the application unless the update includes one or more assemblies that have the same version as existing assemblies. In this case, you must stop the application before installing the update. We recommend, however, that you stop the application in all cases unless you know that the update will not interfere with the application as it runs. For more information, see Application Versioning.

- **When you install multiple .msi files for the same application, there is only one entry made in Add or Remove Programs.** You might do this to update an existing application, for example. You can then use Add or Remove Programs control panel to uninstall the application completely, including any updated items. Note that uninstalling an application by double-clicking the .msi file or using msiexec is not supported. For more information, see How to Uninstall a BizTalk Application.

- **Certificates must be present on all computers that host send ports before an application can run.** The Other People Certificate store contains the certificates used by send ports.

- **If the application includes a virtual directory with the same name as one that already exists on the local computer, resources from the application are added to it.** Otherwise, the virtual directory is created. Any existing files that have the same name as added files will be overwritten. In addition, security settings for an existing virtual directory are not changed, and you should verify that they are sufficiently secure.

- **Create application pools for virtual directories before installing an application.** If your application includes a virtual directory, and the application pool does not already exist in Internet Information Services, you should manually create the application pool before installation. This way, the virtual directory will be bound to the application pool during installation. If you do not create the application, the virtual directory will be bound to the default application pool on installation.

- **You may need to update file names and receive locations for virtual directories before you can install the application.** When installing an application on Internet Information Services (IIS) 5.0 that includes virtual directories for Web services that were created for IIS 6.0, you will need to change any non-ASCII characters in file names to ASCI characters. You will then need to update receive location references accordingly and re-export the .msi file to install. This is because IIS 6.0 fully supports unicode characters in file names, and IIS 5.0 does not. Because changing Web services file names can create errors, we recommend that you always use ASCII characters in file names when you need to move Web services between IIS 6.0 and IIS 5.0.

- **Ensure that BTSHttpReceive.dll is registered as a Web service extension with Internet Information Services (IIS) 6.0.** You must do this if your application includes a virtual directory in order for the HTTP receive location to work.

- **A virtual directory exported from IIS 6.0 may not run correctly if the application is installed on a computer running an earlier version of IIS.** This is because IIS 6.0 does not provide backward compatibility with previous versions. For more information, see your IIS documentation.

- **You may encounter issues when installing an application that includes 64-bit artifacts on a 32-bit computer.** For more information, see How to Add a 64-Bit Artifact to an Application.

- **You should not relocate an installation folder.** Once you install an application, you should not relocate the installation folder or the files it contains. If you do, and then later attempt to remove (uninstall) the application, the removal operation may fail. In particular, the application installation folder contains files generated by BizTalk Server that are necessary to perform the removal. You should not rename, move, or remove these files. The files are:

  - ApplicationDefinition.adf
  - Microsoft.BizTalk.CustomInstaller.dll
  - Microsoft.BizTalk.CustomInstaller.InstallState

To install a BizTalk application

Using the Windows interface

1. Copy the .msi file for the application to the local computer.

2. If you are reinstalling or upgrading an existing BizTalk application, and the new installation includes an assembly that has the same version as one that already exists in the application, or interacts with an artifact that you are updating, ensure that the application is stopped by right-clicking the application folder and then clicking **Stop**.

3. In Windows Explorer, double click the .msi file to start the Installation Wizard.

4. On the Select Installation Folder page, in **Folder**, type the complete installation path for the BizTalk application. Example: C:\Program Files\Generated by BizTalk\MyApplication.

5. Click **Next** four times, and then on the Installation Complete page, click **Close**.

6. If multiple computers will be running the application, repeat the previous steps on each computer.

   Once the application is installed on all computers that will run it, and the application has been imported into the BizTalk group, you can start the application from the BizTalk Server Administration console by right-clicking the application's folder and clicking **Start**. For complete instructions, see How to Start and Stop a BizTalk Application.

Using the command line

1. Copy the .msi file for the application to the local computer.

2. Click **Start**, click **Run**, type **cmd**, and then press Enter.

3. Navigate to the location where the .msi file is stored.

4. Type the following command to install the application, providing the appropriate parameters and values, as shown in the following table:

5. **msiexec** [**/i**] *Package* [**/qn**] **TARGETDIR="***value***"**]

6. Example: **msiexec /i MyApplication.msi**

Parameter	**Value**
/i	Installs the application.
Package	Name of the Windows Installer (.msi) file.

/qn	Performs the installation without displaying a user interface.
TARGETDIR="*value*"	Specifies the application installation folder. This value is also set in the %BTAD_InstallDir% environment variable. Example: TARGETDIR="C:\Programs\BizTalk Applications\My Application"

7. If multiple computers will be running the application, repeat the previous steps on each computer.

   Once the application is installed on all computers that will run it, you can start the application from the BizTalk Server Administration console by right-clicking the application's folder and clicking **Start**. For complete instructions, see How to Start and Stop a BizTalk Application.

Monitoring the Progress of Your BizTalk Application Deployment

You can monitor the progress of your BizTalk application deployment in two ways:

- **BizTalk installation log**: You can consult the installation log that BizTalk Server generates. Installation logs are located in %SystemDrive%\Documents and Settings\<*current user*>\Application Data\Microsoft\BizTalk Server\Deployment.

- **Local event log**: You can track the progress of an installation in the local event log. Therefore, you can track custom installation actions on a per-server basis, or aggregate the actions to a central location using Microsoft Operations Manager (MOM). For more information, see Monitoring BizTalk Server Using MOM.

  You can view the local event log from Event Viewer or the BizTalk Server Administration console. To do the latter, in the left pane of the administration console, click Event Viewer (Local).

- **Windows Installer log**: Microsoft Windows Installer creates a log file on the local computer that logs the actions of a custom action. You can specify this log file by using the /log option of the msiexec command. For more information, see the documentation for Windows Installer.

Customizing Binding Files

Binding files are XML files that describe artifacts in a BizTalk Management database and the relationship between these artifacts. Binding files are useful for exporting configuration information from one BizTalk configuration database and importing this information into another BizTalk configuration database. For example, a developer may design a BizTalk solution and its related artifacts in a development environment, then export these artifacts to a binding file and finally, import these artifacts into a production environment. You can also use a binding file to update an existing configuration. For example you can apply configuration changes made in a development environment to your production environment with a binding file. This topic discusses considerations when updating an existing configuration with a binding file, the structure of a binding file, and adapter specific configuration properties that can be set in a binding file.

In This Section

- Updating an Existing Configuration with a Binding File
- Structure of a Binding File
- Configuration Properties for Integrated BizTalk Adapters

Updating an Existing Configuration with a Binding File

Information in a binding file supersedes existing configuration information. If the name of an artifact in a binding file matches the name of an artifact in your existing configuration, the artifact in the binding file will update the artifact in your existing configuration when you import the binding file.

When updating existing artifacts with binding file artifacts, certain rules are followed. This topic discusses the rules that are followed when updating artifacts in an existing configuration with artifacts in a binding file.

This section assumes that valid values are present in the binding file when the file is imported and does not discuss any scenario where a binding file contains invalid values.

Rules followed by BizTalk Server when updating a configuration with a binding file

BizTalk Server follows certain rules when updating existing artifacts with matching artifacts in a binding file. In general, the following rules are applied:

- Text Boxes and check boxes that are exposed when configuring an artifact via the BizTalk Server user interface (such as the BizTalk Administration Console or BizTalk Explorer), must either be set to a specific value or be empty. Values supplied for artifacts in a binding file will set the user interface value for the updated item accordingly.
- Drop down boxes that are exposed when configuring an artifact via the BizTalk Server user interface must either be set to a specific value or be set to "None". Values supplied for artifacts in a binding file will set the user interface value for the updated item accordingly.
- Datagrid views that are exposed when configuring an artifact via the BizTalk Server user interface are updated with lists from the corresponding item in the

binding file. The list associated with a datagrid view is always overwritten by the list in the binding file unless the datagrid view list is tied to a port or a receive location. In this case the list in the binding file is merged with the existing datagrid view list.

- Artifacts in the binding file are identified by a primary key value. The value associated with the primary key for an artifact can never be set to null in the user interface and so all artifacts in a binding file must have their primary key value set. If the value associated with the primary key for an artifact in the binding file matches the value associated with the primary key for an existing configuration artifact then these artifacts are considered to be identical or matching. If the binding file artifact and the existing artifact are identical then the existing artifact is updated with the binding file artifact as described in the table below. If an artifact in the binding file contains a unique primary key value then a new artifact is created in the BizTalk configuration when the binding file is imported.

This table describes expected behavior if you are updating existing configuration artifacts with matching artifacts by importing a binding file:

Artifact Type	**Property**	**Possible occurrences of specified property**	**User Interface Field**	**Impact of importing matching artifact from binding file.**
Party	Name	Min Occurs: 1 Max Occurs: 1	Text box	Primary key
	Aliases	Min Occurs: 0 Max Occurs: *	Data grid	Overwrite the list of aliases with the list of aliases in the binding file.
	Send Ports	Min Occurs: 0 Max Occurs: *	Data grid	Merge the existing list of ports for this party with the list of ports for this party in the binding file.
	Certificate Common Name and Thumbprint	Min Occurs: 0 Max Occurs: 1 (per property)	Text box	Overwrite these values with the values specified in binding file. If these values do not exist in the binding file then set to null.
Orchestration	Description	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.

	Host	Min Occurs: 0 Max Occurs: 1	Drop down	Overwrite this value with the value specified in binding file. If this value does not exist in the binding file then set to null.
	Inbound ports and outbound ports	Min Occurs: 0 Max Occurs: *	Drop down	Bind a logical port to an existing physical port. The physical port can exist in the following locations: • In the group. • In the application. • In the binding file. Optionally set the port to **None**. If set to **None** then the logical port is not bound to any resource.
	Tracking properties check boxes	Min Occurs: 1 Max Occurs: 1 (per property)	Check box	Overwrite these values with the values specified in the binding file.
Send Port Group	Name	Min Occurs: 1 Max Occurs: 1	Text box	Primary key
	Send ports	Min Occurs: 0 Max Occurs: *	Data grid	Merge the existing list of ports for this send port group with the list of ports for this send port group specified in the binding file.
	Filters	Min Occurs: 0 Max Occurs: *	Data grid	Overwrite the existing list of filters for this send port group with the list of filters for this send port group specified in

				the binding file.
Send Port	Name	Min Occurs: 1 Max Occurs: 1	Text box	Primary key
	Transport - Type	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Transport - Send handler	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Send pipeline	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Retry Count, Retry interval, and Priority	Min Occurs: 1 Max Occurs: 1 (Per property)	Scroll box	Overwrite these values with the values specified in the binding file.
	Ordered delivery	Min Occurs: 1 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file.
	Enable routing for failed messages	Min Occurs: 1 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file.
	Enable Service window	Min Occurs: 1 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file.
	Service window Start time and service window Stop time	Min Occurs: 1 Max Occurs: 1	Scroll box	Overwrite these values with the values specified in the binding file.
	Maps	Min Occurs: 0 Max Occurs: *	Data grid	Overwrite the existing list of maps for this send port with the list of maps for this send port specified in the binding file.

	Filter	Min Occurs: 0 Max Occurs: *	Data grid	Overwrite the existing list of filters for this send port with the list of filters for this send port specified in the binding file.
	Certificate Common Name	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
	Certificate Thumbprint	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
	Tracking	Min Occurs: 0 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file.
	Backup Transport Type	Min Occurs: 0 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Backup Transport URI	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file. Only valid if the backup transport Type is set.
	Backup Transport Send handler	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file. Only valid if the backup transport Type is set.
	Backup Transport Retry count	Min Occurs: 1 Max Occurs: 1	Scroll box	Overwrite this value with the value specified in the binding file. Only valid if the backup transport Type is set.
	Backup Transport Retry interval	Min Occurs: 1 Max Occurs: 1	Scroll box	Overwrite this value with the value specified in the binding file. Only valid if the backup transport Type is set.
	Backup Transport Enable service	Min Occurs: 1	Check box	Overwrite this value with the value specified in the

	window	Max Occurs: 1		binding file. Only valid if the backup transport Type is set.
	Backup Transport Service window Start time and service window Stop time	Min Occurs: 1 Max Occurs: 1	Scroll box	Overwrite these values with the values specified in the binding file. Only valid if the backup transport Type is set and the Enable service window value is set.
Receive Port	Name	Min Occurs: 1 Max Occurs: 1	Text box	Primary key
	Authentication Settings (radio buttons)	Min Occurs: 1 Max Occurs: 1	Radio button	Overwrite this value with the value specified in the binding file.
	Enable Failed Message Routing	Min Occurs: 1 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file.
	Description	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
	Receive Locations	Min Occurs: 0 Max Occurs: *	Data grid	Overwrite the existing list of receive locations for this receive port with the list of receive locations for this receive port specified in the binding file. If all of the receive locations in the binding file already exist in the group then import fails.
	Maps	Min Occurs: 0 Max Occurs: *	Data grid	Overwrite the existing list of maps for this receive port with the list of maps for this receive port specified in the binding file.
	Tracking - Track Message Bodies	Min Occurs: 1	Check box	Overwrite these values with the values specified

	and Track Message Properties	Max Occurs: 1 (per checkbox)		in the binding file.
Receive Location	Name	Min Occurs: 1 Max Occurs: 1	Text box	Primary Key
	Transport type	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Receive Handler	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Pipeline	Min Occurs: 1 Max Occurs: 1	Drop down	Overwrite this value with the value specified in the binding file.
	Description	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
	Schedule Start date and Stop date check boxes and drop down boxes.	Min Occurs: 1 Max Occurs: 1	Check box and drop down box.	Overwrite these values with the values specified in the binding file. Date values are imported even if the check box values are not enabled.
	Enable service window check box	Min Occurs: 1 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file.
	Service window Start time and service window Stop time	Min Occurs: 1 Max Occurs: 1	Scroll box	Overwrite these values with the values specified in the binding file. Only valid if the Enable service window value is set.
Schema	Description	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
	Tracking - Always track all	Min Occurs: 1	Check box	Overwrite this value with the value specified in the

	properties	Max Occurs: 1		binding file.
	Tracking - Select all message properties	Min Occurs: 1 Max Occurs: 1	Check box	Overwrite this value with the value specified in the binding file. If this value is enabled than all message properties that can be checked are also enabled.
	Tracking – individual properties	Min Occurs: 0 Max Occurs: *	Check boxes	Overwrite the existing list of tracked properties for this schema with the list of tracked properties for this schema specified in the binding file. If a binding file is imported which refers to tracked properties that are not available for the existing schema an error is generated.
Map	Description	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
Pipeline	Description	Min Occurs: 1 Max Occurs: 1	Text box	Overwrite this value with the value specified in the binding file.
	Track Events	Min Occurs: 1 Max Occurs: 1 (per checkbox)	Check box	Overwrite these values with the values specified in the binding file.
	Track Message Bodies	Min Occurs: 1 Max Occurs: 1 (per checkbox)	Check box	Overwrite these values with the values specified in the binding file.
Policy	Not applicable. Policies are not exportable to a	Not applicable	Not applicable	Not applicable

	binding file.			
Role Link	Not applicable. Role links are not exportable to a binding file.	Not applicable	Not applicable	Not applicable

Un-binding behavior when updating existing artifacts with matching artifacts in a binding file

Binding file artifacts are typically configured to reference other artifacts, for example a receive port is typically configured to reference a receive location. In this scenario, the receive port is the parent artifact and the receive location is the child artifact. The receive port is **explicitly** configured to reference the receive location and the receive location then **implicitly** references the receive port. If incompletely configured parent artifacts in a binding file exist, for example a receive port that is not configured with a receive location, then they will be incompletely configured after the binding file is imported regardless of their state in the existing configuration. So for example, if you have an existing receive port *myRP* configured with receive location *myRL* and the identical receive port *myRP* in the binding file is **not** configured with receive location *myRL*, then the binding file entry takes precedence. For this example receive port *myRP* will not be configured with a receive location after importing the binding file so you will have effectively **un-bound** *myRL* from *myRP*.

This rule only applies when importing artifacts that make explicit references and not when importing artifacts with implicit references. So if you imported a map that implicitly references (is explicitly referenced by) 10 other artifacts you would not have to be concerned that the map would be un-bound from the implicitly referenced artifacts.

Structure of a Binding File

The structure of binding files exported from the BizTalk Server 2006 Administration Console adheres to a schema that is used to describe all binding files. This topic describes the structure of the binding file and the relationship of certain binding file artifacts to the BizTalk Explorer Object model. This topic should be used as a reference when manually modifying binding files that were exported with the BizTalk Server 2006 Administration Console.

In This Section

- BindingInfo Node

BindingInfo Node

The **BindingInfo** node of a binding file is the root node of a binding file and contains information that applies to all of the entries in the binding file as well as information about the BizTalk Server that the binding file was exported from.

Nodes in the BindingInfo node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Assembly	Attribute	xs:string	Specifies information for the Microsoft.BizTalk.Deployment dll used when creating the binding file. Includes comma separated Version, Culture, and PublicKeyToken attributes for this assembly.	Required	Default value: **"Micros Version=3.0.1.0, PublicKeyToken=31bf3**
Version	Attribute	xs:string	Specifies the version of BizTalk Server that the binding file was generated on.	Required	Default value: **3.5.1.0**
BindingStatus	Attribute	BindingState (SimpleType)	Specifies the binding status of the artifacts exported with the binding file.	Required	Default value: None Valid values: • Unknown • NoBindings • Unbound • PartiallyBound • FullyBound
BoundEndpoints	Attribute	xs:int	Specifies the number of bound endpoints in the binding file.	Required	Default value: none
TotalEndpoints	Attribute	xs:int	Specifies the total number of endpoints in the binding file.	Required	Default value: none

Description	Element	xs:string	Specifies a text description of the BindingInfo section of the binding file.	Not required	Default value: empty
Timestamp	Element	xs:dateTime	Specifies when the binding file was exported.	Required	Default value: Time on th binding file was exported.
ModuleRefCollection Node	Record	ArrayOfModuleRef (Complex Type)	Container node for the .NET assemblies exported with the binding file.	Not required	Default value: none
SendPortCollection Node	Record	ArrayOfSendPort (Complex Type)	Container node for the send ports exported with the binding file.	Not required	Default value: none
DistributionList Collection Node	Record	ArrayOfDistributionList (Complex Type)	Container node for the distribution lists exported with the binding file.	Not required	Default value: none
ReceivePortCollection Node	Record	ArrayOfReceivePort (Complex Type)	Container node for the receive ports exported with the binding file.	Not required	Default value: none
PartyCollection Node	Record	ArrayOfParty (Complex Type)	Container node for the parties exported with the binding file.	Not required	Default value: none

ModuleRefCollection Node

The ModuleRefCollection section of a binding file is the parent node for all of the ModuleRef nodes which contain specific information about .NET assemblies exported with the binding file.

Entries in the ModuleRefCollection section

The following table lists the properties that can be set for the nodes in this section of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments

ModuleRef (ModuleRefCollection Node)	Record	ModuleRef (ComplexType)	Container node for a .NET assembly module exported with the binding file.	Not required	Default value: None

ModuleRef (ModuleRefCollection Node)

The ModuleRef node of a binding file contains specific information about a .NET assembly that was exported with the binding file. A ModuleRef node can include but is not limited to descriptions of assemblies that contain custom code, schemas, and orchestrations.

Nodes in the ModuleRef node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Services (ModuleRef Node)	Record	ArrayOfServiceRef (ComplexType)	Container node for services associated with this .NET assembly.	Not required	Default value: none
TrackedSchemas (ModuleRef Node)	Record	ArrayOfSchema (ComplexType)	Container node for schemas associated with this .NET assembly	Not required	Default value: none

Services (ModuleRef Node)

The Services node of a binding file contains specific information about each service that is exported with the binding file.

Nodes in the Services node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Service (Services	Record	ServiceRef (ComplexType)	Container node for nodes that describe	Not required	Default value: None

Node)			this service.		

Service (Services Node)

The Service node of a binding file contains information about a service exported with the binding file. The service node also contains nodes that describe the ports and roles associated with the service and a node that describes the host associated with the service.

Nodes in the Service node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the service.	Required	Default value: empty
State	Attribute	ServiceRefState (SimpleType)	Specifies the state of the service.	Required	Default value: Default Possible values include: • Default • Unenlisted • Enlisted • Started
TrackingOption	Attribute	OrchestrationTrackingTypes (SimpleType)	Specifies the message tracking options for the service.	Required	Default value: none Possible values inclu those available in th OrchestrationTrackingType enumeration.
Description	Attribute	xs:string	Specifies a description for the service.	Not required	Default value: empty
Ports (Service Node)	Record	ArrayOfServicePortRef (ComplexType)	Container node for the ports bound	Not required	Default value: none

			to the service.		
Roles (Service Node)	Record	ArrayOfRoleRef (ComplexType)	Container node for the roles bound to the service.	Not required	Default value: none
Host (Service Node)	Record	HostRef (ComplexType)	Container node for the host bound to the service.	Required	Default value: none

Ports (Service Node)

The Ports node of a binding file is the parent node for all of the Port nodes which contain specific information about ports bound to a service that is exported with the binding file.

Node in the Ports node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Port (Ports Node)	Record	ServicePortRef (ComplexType)	Specifies information about a port that is bound to a service that is exported with the binding file.	Not required	Default value: none

Port (Ports Node)

The port node of a binding file contains specific information about a port or distribution list that is bound to a service that is exported with the binding file.

Nodes in the Port node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the port.	Not required	Default value: empty
Modifier	Attribute	xs:int	Specifies the type modifier for the port.	Required	Default value: none Possible values include those available in the PortModifier enumeration.
BindingOption	Attribute	xs:int	Defines the type of binding for the port.	Required	Default value: none Possible values include those available in the BindingType enumeration.
SendPortRef (Port Node)	Record	SendPortRef (ComplexType)	Container node for send ports that is referenced by a service.	Not required	Default value: empty
DistributionListRef (Port Node)	Record	DistributionListRef (ComplexType)	Container node for distribution lists referenced by a service.	Not required	Default value: empty
ReceivePortRef (Port Node)	Record	ReceivePortRef (ComplexType)	Container node for receive ports referenced by a service.	Not required	Default value: emp

Roles (Service Node)

The Roles node of a binding file is the parent node for all of the Role nodes which provide specific information about each role bound to a service that is exported with the binding file.

Nodes in the Roles node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Role (Roles Node)	Record	RoleRef (ComplexType)	Specifies information about a role that is bound to a service that is exported with the binding file.	Not required	Default value: none

Role (Roles Node)

The Role node of the Roles node of a binding file specifies information about a role that is bound to a service that is exported with the binding file.

Nodes in the Role node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the role.	Not required	Default value: empty
RoleLinkTypeName	Attribute	xs:string	Specifies the name of the role link type associated with the role	Not required	Default value: empty
RoleType	Attribute	RoleRefType (SimpleType)	Specifies the role type associated with the role.	Required	Default value: none Possible values include:

					• Unknown • Implements • Uses
Enlisted Parties (Role Node)	Record	ArrayOfEnlistedParty (ComplexType)	Container node for the enlisted parties bound to this role.	Not required	Default value: none

Host (Service Node)

The Host node of the Service node of a binding file describes the host associated with the service that is exported with the binding file.

Nodes in the Host node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the host.	Not required	Default value: empty
NTGroupName	Attribute	xs:string	Specifies the Windows NT Group name associated with the host.	Not required	Default value: empty
Type	Attribute	xs:int	Specifies the host type as in process or isolated.	Required	Default value: none Possible values are described in the HostType enumeration.
Trusted	Attribute	xs:boolean	Specifies whether the BizTalk host can be trusted to collect authentication information.	Required	Default value: none Set to **true** if the host is trusted, otherwise set to

					false.

TrackedSchemas (ModuleRef Node)

The TrackedSchemas node of a binding file is the parent node for all of the Schema nodes which describe the schemas bound to the service that is exported with the binding file.

Nodes in the TrackedSchemas node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Schema (TrackedSchemas Node)	Record	ArrayOfSchema (ComplexType)	Container node for the schemas that are bound to the service that is exported with the binding file.	Not required	Default value: none

Schema (TrackedSchemas Node)

The Schema node of the TrackedSchemas node of a binding file describes a schema bound to a service that is exported with the binding file.

Nodes in the Schema node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
FullName	Attribute	xs:string	Specifies the full name for the schema.	Not required	Default value: empty
AssemblyQualifiedName	Attribute	xs:string	Specifies the qualified name for the assembly containing this schema.	Not required	Default value: empty

AlwaysTrackAllProperties	Attribute	xs:boolean	Specifies whether to track all properties for the specified assembly.	Required	Default value: none Set to **true** to track all properties, otherwise set to **false**.
Description	Attribute	xs:string	Specifies a description for the schema.	Not required	Default value: empty
TrackedPropertyNames (Schema Node)	Record	ArrayOfString (ComplexType)	Container for the elements that specify the properties to be tracked.	Not required	Default value: none

TrackedPropertyNames (Schema Node)

The TrackedPropertyNames node of a binding file contains elements that specify which properties in the parent schema are to be tracked.

Nodes in the TrackedPropertyName node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
string	Element	xs:string	Specifies a property name that will be tracked.	Not required	Default value: empty

SendPortCollection Node

The SendPortCollection node of a binding file is the parent node for all of the SendPort nodes which contain specific information about a send port that is exported with the binding file.

Nodes in the SendPortCollection node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
SendPort (SendPortCollection Node)	Record	SendPort (ComplexType)	Specifies information about a send port that is exported with the binding file.	Not required	Default value: none

SendPort (SendPortCollection Node)

The SendPort node of a binding file contains specific information about a send port that is exported with the binding file.

Nodes in the SendPort node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the send port.	Not required	Default value: empty
IsStatic	Attribute	xs:boolean	Specifies whether the send port is static or dynamic.	Required	Default value: none
IsTwoWay	Attribute	xs:boolean	Specifies whether the send port is one way or is solicit-response (two way).	Required	Default value: none Possible values are d **MSBTS_SendPort.IsTwoW (WMI)**
BindingOption	Attribute	xs:int	Specifies the type of binding for the orchestration	Required	Default value: none Possible values are docu BindingType enumeration.

			port.		
Description	Element	xs:string	Specifies a description for the send port.	Required	Default value: empty
TransmitPipeline (SendPort Node)	Record	PipelineRef (ComplexType)	Specifies the send pipeline associated with the send port.	Not required	Default value: none
SendPipelineData	Element	xs:string	Specifies the custom configuration specific to this instance of the usage of the pipeline.	Not required	Default value: empty.
PrimaryTransport (SendPort Node)	Record	TransportInfo (ComplexType)	Specifies information about the primary transport that the send port is configured to use.	Not required	Default value: none
SecondaryTransport (SendPort Node)	Record	TransportInfo (ComplexType)	Specifies information about the secondary transport that the send port is configured to use.	Not required	Default value: none
EncryptionCert (SendPort Node)	Record	CertificateInfo (ComplexType)	Specifies information about the encryption certificate used with the send	Not required	Default value: none

			port.		
ReceivePipeline (SendPort Node)	Record	PipelineRef (ComplexType)	Specifies information about any receive pipelines used with the send port.	Not required	Default value: none
ReceivePipelineData	Element	xs:string	Specifies the custom configuration specific to this instance of the usage of the pipeline.	Required	Default value: empty
Tracking	Element	xs:int	Specifies the level of document tracking for the send port	Required	Default value: none Possible values are docu TrackingTypes enumeration.
Filter	Element	xs:string	Specifies the name of the optional filter expression used on this send port.	Required	Default value: empty Possible values are d **MSBTS_SendPort.Filter Pr**
Transforms (SendPort Node)	Record	ArrayOfTransform (ComplexType)	Specifies the collection of outbound transforms of a one way send port.	Not required	Default value: none
InboundTransforms (SendPort Node)	Record	ArrayOfTransform (ComplexType)	Specifies the collection of inbound transforms of a two-way send port.	Not required	Default value: none

OrderedDelivery	Element	xs:boolean	Specifies whether or not the send port orders the delivery of messages.	Required	Default value: none Possible values are d **MSBTS_SendPort.Ordered Property (WMI)**
Priority	Element	xs:int	Specifies the priority of the send port.	Required	Default value: 5 Possible values are d **MSBTS_SendPort.Priority (WMI)**
StopSendingOnFailure	Element	xs:boolean	Specifies whether or not the send port stops sending messages on a failure.	Required	Default value: none Possible values are d **MSBTS_SendPort.StopSer Property (WMI)**
RouteFailedMessage	Element	xs:boolean	Specifies whether or not failed messages are routed to failed message subscribers.	Required	Default value: none Possible values are d **MSBTS_SendPort.RouteFa Property (WMI)**
ApplicationName	Element	xs:string	Specifies the name of the application associated with the send port.	Required	Default value: empty Possible values are d **ISSOMapping.Application Property (COM)**

TransmitPipeline (SendPort Node)

The TransmitPipeline node of the SendPort node of a binding file provides specific information about the send pipeline bound to a send port exported with a binding file.

Nodes in the TransmitPipeline node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the send pipeline.	Not required	Default value: empty
FullyQualifiedName	Attribute	xs:string	Specifies the fully qualified name of the pipeline, which includes the name of the assembly that the pipeline was deployed as a part of	Not required	Default value: empty
Type	Attribute	xs:int	Specifies the type of pipeline.	Required	Default value: none Possible values are documented in the PipelineType enumeration.
TrackingOption	Attribute	PipelineTrackingTypes (SimpleType)	Specifies the tracking options for the pipeline.	Required	Default value: none Possible values are documented in the PipelineTrackingTypes enumeration.
Description	Attribute	xs:string	Specifies a description for the send pipeline.	Not required	Default value: empty

PrimaryTransport (SendPort Node)

The PrimaryTransport node of the SendPort node of a binding file provides specific information about the primary transport that is bound to a send port exported with the binding file.

Nodes in the PrimaryTransport node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Address	Element	xs:string	Specifies the address (or URI) of the transport.	Not required	Default value: empty
TransportType (PrimaryTransport Node)	Record	ProtocolType (ComplexType)	Specifies the transport type, which is also the name of the adapter used for this transport.	Not required	Default value: none
TransportTypeData	Element	xs:string	Specifies configuration information specific to the adapter.	Not required	Default value: empty See Configuration Properties for Integrated BizTalk Adapters for adapter specific information about the properties that can be stored in this string.
RetryCount	Element	xs:int	Specifies the retry count for the adapter used with the transport.	Required	Default value: none
RetryInterval	Element	xs:int	Specifies the retry interval	Required	Default

			in minutes for the adapter used with the transport.		value: none
ServiceWindowEnabled	Element	xs:boolean	Specifies whether the service window is enabled for the adapter used with the transport.	Required	Default value: none Set to **true** if service window is enabled, otherwise set to **false**.
FromTime	Element	xs:dateTime	Specifies the start time for the service window.	Required	Default value: none
ToTime	Element	xs:dateTime	Specifies the end time for the service window.	Required	Default value: none
Primary	Element	xs:boolean	Specifies whether the adapter used with the transport is primary.	Required	Default value: none Set to **true** if the adapter used with the transport is primary, otherwise set to **false**.
OrderedDelivery	Element	xs:boolean	Specifies whether or not the adapter used with the transport should send messages in an ordered	Required	Default value: none Set to **true** if the transport is to send messages in order, otherwise set

			manner.		to **false**.
DeliveryNotification	Element	xs:int	Specifies whether or not the adapter used with the transport should return a delivery notification indicating if the transmission was successful.	Required	Default value: none Set to **true** for delivery notifications, otherwise set to **false**.
SendHandler (PrimaryTransport Node)	Record	SendHandlerRef (ComplexType)	Specifies the send handler for the adapter used with the transport.	Required	Default value: none

TransportType (PrimaryTransport Node)

The TransportType node of the PrimaryTransport node of a binding file contains specific information about the adapter associated with a transport that is exported with the binding file.

Nodes in the TransportType node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the adapter associated with the transport.	Not required	Default value: empty
Capabilities	Attribute	xs:int	Specifies the capabilities of the adapter	Required	Default value: none

			associated with the transport.		Possible values include those available in the Capabilities enumeration.
ConfigurationClsid	Attribute	xs:string	Specifies the configuration GUID of the adapter associated with the transport.	Not required	Default value: empty

SendHandler (PrimaryTransport Node)

The SendHandler node of the PrimaryTransport node of a binding file contains specific information about the send handler associated with a transport that is exported with the binding file.

Nodes in the SendHandler node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the send handler associated with the transport.	Not required	Default value: empty
HostTrusted	Attribute	xs:boolean	Specifies whether the host associated with the send handler is trusted.	Required	Default value: none Set to **true** if host is trusted, otherwise set to **false**.
TransportType	Record	ProtocolType (ComplexType)	Specifies the transport type, which is also the name of the adapter used with this send handler.	Required	Default value:

TransportType

The TransportType node of the SendHandler node of a binding file contains specific information about the adapter associated with a send handler that is exported with the binding file.

Nodes in the TransportType node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the adapter associated with the send handler.	Not required	Default value: empty
Capabilities	Attribute	xs:int	Specifies the capabilities of the adapter associated with the send handler.	Required	Default value: none Possible values include those available in the Capabilities enumeration.
ConfigurationClsid	Attribute	xs:string	Specifies the configuration GUID of the adapter associated with the send handler.	Not required	Default value: empty

SecondaryTransport (SendPort Node)

The SecondaryTransport node of the SendPort node of a binding file provides specific information about the secondary transport that is bound to a send port exported with the binding file. If a secondary transport is specified, it is used when all retry attempts with the primary transport have been exhausted.

Nodes in the SecondaryTransport node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Address	Element	xs:string	Specifies the address (or URI) of the transport.	Not required	Default value: empty
TransportType (SecondaryTransport Node)	Record	ProtocolType (ComplexType)	Specifies the transport type, which is also the name of the adapter used for this transport.	Not required	Default value: none
TransportTypeData	Element	xs:string	Specifies configuration information specific to the adapter.	Not required	Default value: empty See Configuration Properties for Integrated BizTalk Adapters for adapter specific information about the properties that can be stored in this string.
RetryCount	Element	xs:int	Specifies the retry count for the adapter used with the transport.	Required	Default value: none
RetryInterval	Element	xs:int	Specifies the retry interval in minutes for the adapter used with	Required	Default value: none

			the transport.		
ServiceWindowEnabled	Element	xs:boolean	Specifies whether the service window is enabled for the adapter used with the transport.	Required	Default value: none Set to **true** if service window is enabled, otherwise set to **false**.
FromTime	Element	xs:dateTime	Specifies the start time for the service window.	Required	Default value: none
ToTime	Element	xs:dateTime	Specifies the end time for the service window.	Required	Default value: none
Primary	Element	xs:boolean	Specifies whether the adapter used with the transport is primary.	Required	Default value: none Set to **true** if the adapter used with the transport is primary, otherwise set to **false**.
OrderedDelivery	Element	xs:boolean	Specifies whether or not the adapter used with the transport should send messages in an ordered manner.	Required	Default value: none Set to **true** if the transport is to send messages in order, otherwise set to **false**.
DeliveryNotification	Element	xs:int	Specifies whether or	Required	Default

			not the adapter used with the transport should return a delivery notification indicating if the transmission was successful.		value: none Set to **true** for delivery notifications, otherwise set to **false**.
SendHandler (SecondaryTransport Node)	Record	SendHandlerRef (ComplexType)	Specify the send handler for the adapter used with the transport.	Required	Default value: none

TransportType (SecondaryTransport Node)

The TransportType node of the SecondaryTransport node of a binding file contains specific information about the adapter associated with a transport that is exported with the binding file.

Nodes in the TransportType node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the adapter associated with the transport.	Not required	Default value: empty
Capabilities	Attribute	xs:int	Specifies the capabilities of the adapter associated with the transport.	Required	Default value: none Possible values include those available in the

					Capabilities enumeration.
ConfigurationClsid	Attribute	xs:string	Specifies the configuration GUID of the adapter associated with the transport.	Not required	Default value: empty

SendHandler (SecondaryTransport Node)

The SendHandler node of the SecondaryTransport node of a binding file contains specific information about the send handler associated with a transport that is exported with the binding file.

Nodes in the SendHandler node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the send handler associated with the transport.	Not required	Default value: empty
HostTrusted	Attribute	xs:boolean	Specifies whether the host associated with the send handler is trusted.	Required	Default value: none Set to **true** if host is trusted, otherwise set to **false**.
TransportType (SendHandler Node)	Record	ProtocolType (ComplexType)	Specifies the transport type, which is also the name of the adapter used with this send handler.	Required	Default value:

TransportType (SendHandler Node)

The TransportType node of the SendHandler node of a binding file contains specific information about the adapter associated with a send handler that is exported with the binding file.

Nodes in the TransportType node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the adapter associated with the send handler.	Not required	Default value: empty
Capabilities	Attribute	xs:int	Specifies the capabilities of the adapter associated with the send handler.	Required	Default value: none Possible values include those available in the Capabilities enumeration.
ConfigurationClsid	Attribute	xs:string	Specifies the configuration GUID of the adapter associated with the send handler.	Not required	Default value: empty

EncryptionCert (SendPort Node)

The EncryptionCert node of the SendPort node of a binding file contains information about the encryption certificate used with a send port that is exported with the binding file.

Nodes in the EncryptionCert node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
LongName	Attribute	xs:string	Specifies the long name of the certificate.	Not required	Default value: empty
ShortName	Attribute	xs:string	Specifies the short name of the certificate.	Not required	Default value: empty
UsageType	Attribute	CertUsageType (SimpleType)	Specifies the intended usage of this certificate	Required	Default value: none Possible values include those available in the CertUsageType enumeration.
ThumbPrint	Attribute	xs:string	Specifies the thumbprint, or unique ID, of the certificate.	Not required	Default value: empty

ReceivePipeline (SendPort Node)

The ReceivePipeline node of the SendPort node of a binding file contains specific information about a receive pipeline bound to a two way send port that is exported with the binding file.

Nodes in the ReceivePipeline node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the send pipeline.	Not required	Default value: empty
FullyQualifiedName	Attribute	xs:string	Specifies the fully qualified name of the pipeline, which includes the	Not required	Default value: empty

			name of the assembly that the pipeline was deployed as a part of.		
Type	Attribute	xs:int	Specifies the type of pipeline.	Required	Default value: none Possible values are documented in the PipelineType enumeration.
TrackingOption	Attribute	PipelineTrackingTypes (SimpleType)	Specifies the tracking options for the pipeline.	Required	Default value: none Possible values are documented in the PipelineTrackingTypes enumeration.
Description	Attribute	xs:string	Specifies a description for the send pipeline.	Not required	Default value: empty

Transforms (SendPort Node)

The Transforms node of the SendPort node of a binding file contains the collection of outbound transforms of a one way send port that is exported with the binding file.

Nodes in the Transforms node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Transform (SendPort\Transforms Node)	Record	Transform (ComplexType)	Specifies a BizTalk Server map, or transform, which is an item that represents the mapping between a	Not required	Default value: none

			source schema and destination schema.		

Transform (SendPort\Transforms Node)

The Transform node of the Transforms node of a binding file contains specific information about a BizTalk Server map that is exported with the binding file.

Nodes in the Transform node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
FullName	Attribute	xs:string	Specifies the full name of the map.	Not required	Default value: empty
AssemblyQualifiedName	Attribute	xs:string	Specifies the assembly qualified name of the map.	Not required	Default value: empty

InboundTransforms (SendPort Node)

The InboundTransforms node of the SendPort node of a binding file contains the collection of inbound transforms of a two-way send port that is exported with the binding file.

Nodes in the InboundTransforms node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Transform (InboundTransforms Node)	Record	Transform (ComplexType)	Specifies a BizTalk Server map, or transform, which is an item that represents the mapping between a source schema and destination	Not required	Default value: none

			schema.		

Transform (InboundTransforms Node)

The Transform node of the InboundTransforms node of a binding file contains specific information about a BizTalk Server map that is exported with the binding file.

Nodes in the Transform node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
FullName	Attribute	xs:string	Specifies the full name of the map.	Not required	Default value: empty
AssemblyQualifiedName	Attribute	xs:string	Specifies the assembly qualified name of the map.	Not required	Default value: empty

DistributionListCollection Node

The DistributionListCollection node of a binding file is the parent node for all of the DistributionList nodes which contain specific information about distribution lists that are exported with the binding file.

Nodes in the DistributionListCollection node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
DistributionList (DistributionListCollection Node)	Record	DistributionList (ComplexType)	Specifies information about a distribution list that is exported with the binding file.	Not required	Default value: none

DistributionList (DistributionListCollection Node)

The DistributionList node of a binding file contains specific information about a distribution list that is exported with the binding file. A distribution list is referred to as a send port group in the BizTalk Server Administrator.

Nodes in the DistributionList node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the distribution list.	Not required	Default value: empty
SendPorts (DistributionList Node)	Record	ArrayOfSendPortRef (ComplexType)	Specifies the send port or send ports included in the distribution list.	Not required	Default value: none
Filter	Element	xs:string	Specifies the name of the optional filter expression used on this distribution list.	Required	Default value: empty
ApplicationName	Element	xs:string	Specifies the name of the application that the distribution list is associated with.	Required	Default value: empty
Description	Element	xs:string	Specifies a description for the distribution list.	Required	Default value: empty

SendPorts (DistributionList Node)

The SendPorts node of the DistributionList node of a binding file is the container node for the send port references in the distribution list.

Nodes in the SendPorts node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
SendPortRef (SendPorts Node)	Record	SendPortRef (ComplexType)	Container node for a reference to a send port made by the distribution list.	Not required	Default value: none

SendPortRef (SendPorts Node)

The SendPortRef node of the SendPorts node of a binding file specifies the name of a send port referenced by a distribution list.

Nodes in the SendPortRef node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the send port referenced by the distribution list.	Not required	Default value: empty

ReceivePortCollection Node

The ReceivePortCollection node of a binding file is the parent node for all of the ReceivePort nodes which contain specific information about a receive port that is exported with the binding file.

Nodes in the ReceivePortCollection node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
ReceivePort (ReceivePortCollection Node)	Record	ReceivePort (ComplexType)	Specifies information about a receive port that is exported with the binding file.	Not required	Default value: none

ReceivePort (ReceivePortCollection Node)

The ReceivePort node of the ReceivePortCollection node of a binding file contains specific information about a receive port that is exported with the binding file.

Nodes in the ReceivePort node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the receive port.	Not required	Default value: empty
IsTwoWay	Attribute	xs:boolean	Specifies whether the receive port is one way or is request-response (two way).	Required	Default value: none Possible values are **MSBTS_SendPort.IsT** **(WMI)**
BindingOption	Attribute	xs:int	Specifies the type of binding for the orchestration port.	Required	Default value: none Possible values are c BindingType enumerat
Description	Element	xs:string	Specifies a description for the receive port.	Required	Default value: empty

ReceiveLocations (ReceivePort Node)	Record	ArrayOfReceiveLocation (ComplexType)	Container node for the receive locations associated with this receive port.	Not required.	Default value: none
TransmitPipeline (ReceivePort Node)	Record	PipelineRef (ComplexType)	Specifies the send pipeline associated with the receive port if the receive port is a two way receive port.	Not required	Default value: none
SendPipelineData	Element	xs:string	Specifies the custom configuration specific to this instance of the usage of the pipeline.	Not required	Default value: empty.
Authentication	Element	xs:int	Specifies an enumeration value indicating whether authentication is needed at this receive port.	Required	Default value: none Possible values are o AuthenticationType enu
Tracking	Element	xs:int	Specifies the level of document tracking for the receive port	Required	Default value: none Possible values are o TrackingTypes enumer
Transforms (ReceivePort Node)	Record	ArrayOfTransform (ComplexType)	Specifies the collection of inbound transforms of a one way	Not required	Default value: none

			receive port.		
OutboundTransforms (ReceivePort Node)	Record	ArrayOfTransform (ComplexType)	Specifies the collection of outbound transforms to apply to documents on a two-way receive port	Not required	Default value: none
RouteFailedMessage	Element	xs:boolean	Specifies whether or not failed messages are routed to failed message subscribers.	Required	Default value: none Possible values are **MSBTS_SendPort.Ro Property (WMI)**
ApplicationName	Element	xs:string	Specifies the name of the application associated with the receive port.	Required	Default value: empty Possible values are **ISSOMapping.Applic Property (COM)**

ReceiveLocations (ReceivePort Node)

The ReceiveLocations node of the ReceivePort node of a binding file is the parent node for all of the ReceiveLocation nodes which contain specific information about the receive locations that are exported with the binding file.

Nodes in the ReceiveLocations node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
ReceiveLocation (ReceiveLocations Node)	Record	ReceiveLocation (ComplexType)	Specifies information about a receive location that is exported with the binding file.	Not required	Default value: none

ReceiveLocation (ReceiveLocations Node)

The ReceiveLocation node of the ReceiveLocations node of a binding file contains specific information about a receive location that is exported with the binding file.

Nodes in the ReceiveLocation node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the receive location.	Not required	Default value: empty
Description	Element	xs:string	Specifies a description for the receive location.	Required	Default value: empty
Address	Element	xs:string	Specifies the address of the receive location.	Not required	Default value: empty
PublicAddress	Element	xs:string	Specifies the public address of the receive location.	Required	Default value: empty
Primary	Element	xs:boolean	Specifies whether the receive location is primary.	Required	Default value: none
ReceiveLocationServiceWindowEnabled	Element	xs:boolean	Specifies whether the service window is enabled.	Required	Default value: none Specify **true** if the servi window is enabled; otherwis specify **false.**
ReceiveLocationFromTime	Element	xs:dateTime	Specifies the start time of the service window.	Required	Default value: none
ReceiveLocationToTime	Element	xs:dateTime	Specifies the end time of the service window.	Required	Default value: none
ReceiveLocationStartDateEnabled	Element	xs:boolean	Specifies whether the start date for the service window	Required	Default value: none

			is enabled.		
ReceiveLocationStartDate	Element	xs:dateTime	Specifies the start date of the service window.	Required	Default value: none
ReceiveLocationEndDateEnabled	Element	xs:boolean	Specifies whether the end date for the service window is enabled.	Required	Default value: none
ReceiveLocationEndDate	Element	xs:dateTime	Specifies the end date of the service window.	Required	Default value: none
ReceiveLocationTransportType (ReceiveLocation Node)	Record	ProtocolType (ComplexType)	Specifies the transport type for this receive location	Required	Default value: none
ReceiveLocationTransportTypeData	Element	xs:string	Specifies the transport type properties for the receive location.	Not required	Default value: empty See Configuration Properti for Integrated BizTa Adapters for adapter speci information about t properties that can be stor in this string.
ReceivePipeline (ReceiveLocation Node)	Record	PipelineRef (ComplexType)	Specifies the receive pipeline for the receive location.	Required	Default value: none
ReceivePipelineData	Element	xs:string	Specifies the custom configuration specific to the receive pipeline used for this receive location.	Required	Default value: empty
SendPipeline (ReceiveLocation Node)	Record	PipelineRef (ComplexType)	Specifies the send pipeline for a two way receive location.	Required	Default value: none

SendPipelineData	Element	xs:string	Specifies the custom configuration specific to the send pipeline used for this receive location.	Required	Default value: empty
EncryptionCert (ReceiveLocation Node)	Record	CertificateInfo (ComplexType)	Specifies the encryption certificate associated with the receive location.	Not required	Default value: none
Enable	Element	xs:boolean	Specifies whether the receive location is enabled or not.	Required	Default value: none
ReceiveHandler (ReceiveLocation Node)	Record	ReceiveHandlerRef (ComplexType)	Specifies the receive handler to use for this receive location.	Not required	Default value: none

ReceiveLocationTransportType (ReceiveLocation Node)

The ReceiveLocationTransportType node of the ReceiveLocation node of a binding file contains specific information about the adapter associated with a transport that is exported with the binding file.

Nodes in the ReceiveLocationTransportType node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the adapter associated with the transport.	Not required	Default value: empty
Capabilities	Attribute	xs:int	Specifies the capabilities of the adapter associated with the transport.	Required	Default value: none Possible values include those

					available in the Capabilities enumeration.
ConfigurationClsid	Attribute	xs:string	Specifies the configuration GUID of the adapter associated with the transport.	Not required	Default value: empty

ReceivePipeline (ReceiveLocation Node)

The ReceivePipeline node of the ReceiveLocation node of a binding file contains specific information about a receive pipeline used with a receive location that is exported with the binding file.

Nodes in the ReceivePipeline node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the receive pipeline.	Not required	Default value: empty
FullyQualifiedName	Attribute	xs:string	Specifies the fully qualified name of the pipeline, which includes the name of the assembly that the pipeline was deployed as	Not required	Default value: empty

			a part of		
Type	Attribute	xs:int	Specifies the type of pipeline.	Required	Default value: none Possible values are documented in the PipelineType enumeration.
TrackingOption	Attribute	PipelineTrackingTypes (SimpleType)	Specifies the tracking options for the pipeline.	Required	Default value: none Possible values are documented in the PipelineTrackingTypes enumeration.
Description	Attribute	xs:string	Specifies a description for the receive pipeline.	Not Required	Default value: empty

SendPipeline (ReceiveLocation Node)

The SendPipeline node of the ReceiveLocation node of a binding file provides specific information about the send pipeline bound to a receive location that is exported with the binding file.

Nodes in the SendPipeline node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the send pipeline.	Not required	Default value: empty
FullyQualifiedName	Attribute	xs:string	Specifies the fully qualified name of the pipeline, which includes the	Not required	Default value: empty

			name of the assembly that the pipeline was deployed as a part of		
Type	Attribute	xs:int	Specifies the type of pipeline.	Required	Default value: none Possible values are documented in the PipelineType enumeration.
TrackingOption	Attribute	PipelineTrackingTypes (SimpleType)	Specifies the tracking options for the pipeline.	Required	Default value: none Possible values are documented in the PipelineTrackingTypes enumeration.
Description	Attribute	xs:string	Specifies a description for the send pipeline.	Not required	Default value: empty

EncryptionCert (ReceiveLocation Node)

The EncryptionCert node of the ReceiveLocation node of a binding file contains information about the encryption certificate used with a receive location that is exported with the binding file.

Nodes in the EncryptionCert node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
LongName	Attribute	xs:string	Specifies the long name of the certificate.	Not required	Default value: empty
ShortName	Attribute	xs:string	Specifies the short name of the certificate.	Not required	Default value: empty

UsageType	Attribute	CertUsageType (SimpleType)	Specifies the intended usage of this certificate	Required	Default value: none Possible values include those available in the CertUsageType enumeration.
ThumbPrint	Attribute	xs:string	Specifies the thumbprint, or unique ID, of the certificate.	Not required	Default value: empty

ReceiveHandler (ReceiveLocation Node)

The ReceiveHandler node of the ReceiveLocation node of a binding file contains specific information about the receive handler associated with a transport that is exported with the binding file.

Nodes in the ReceiveHandler node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the receive handler associated with the transport.	Not required	Default value: empty
HostTrusted	Attribute	xs:boolean	Specifies whether the host associated with the receive handler is trusted.	Required	Default value: none Set to **true** if host is trusted, otherwise set to **false**.
TransportType (ReceiveHandler Node)	Record	ProtocolType (ComplexType)	Specifies the transport type, which is also the name of the adapter used with this receive	Required	Default value: none

			handler.		

TransportType (ReceiveHandler Node)

The TransportType node of the ReceiveHandler node of a binding file contains specific information about the adapter associated with a receive handler that is exported with the binding file.

Nodes in the TransportType node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the adapter associated with the receive handler.	Not Required	Default value: empty
Capabilities	Attribute	xs:int	Specifies the capabilities of the adapter associated with the receive handler.	Required	Default value: none Possible values include those available in the Capabilities enumeration.
ConfigurationClsid	Attribute	xs:string	Specifies the configuration GUID of the adapter associated with the receive handler.	Not Required	Default value: empty

TransmitPipeline (ReceivePort Node)

The TransmitPipeline node of the ReceivePort node of a binding file provides specific information about the send pipeline bound to a two way receive port exported with the binding file.

Nodes in the TransmitPipeline node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
Name	Attribute	xs:string	Specifies the name of the send pipeline.	Not required	Default value: empty
FullyQualifiedName	Attribute	xs:string	Specifies the fully qualified name of the pipeline, which includes the name of the assembly that the pipeline was deployed as a part of.	Not required	Default value: empty
Type	Attribute	xs:int	Specifies the type of pipeline.	Required	Default value: none Possible values are documented in the PipelineType enumeration.
TrackingOption	Attribute	PipelineTrackingTypes (SimpleType)	Specifies the tracking options for the pipeline.	Required	Default value: none Possible values are documented in the PipelineTrackingTypes enumeration.
Description	Attribute	xs:string	Specifies a description for the send pipeline.	Not required	Default value: empty

Transforms (ReceivePort Node)

The Transforms node of the ReceivePort node of a binding file contains the collection of inbound transforms of a one way receive port that is exported with the binding file.

Nodes in the Transforms node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Transform (Transforms Node)	Record	Transform (ComplexType)	Specifies a BizTalk Server map, or transform, which is an item that represents the mapping between a source schema and destination schema.	Not required	Default value: none

Transform (Transforms Node)

The Transform node of the Transforms node of a binding file contains specific information about a BizTalk Server map that is exported with the binding file.

Nodes in the Transform node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
FullName	Attribute	xs:string	Specifies the full name of the map.	Not required	Default value: empty
AssemblyQualifiedName	Attribute	xs:string	Specifies the assembly qualified name of the map.	Not required	Default value: empty

OutboundTransforms (ReceivePort Node)

The OutboundTransforms node of the ReceivePort node of a binding file contains the collection of outbound transforms of a two-way receive port that is exported with the binding file.

Nodes in the OutboundTransforms node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Transform (OutboundTransforms Node)	Record	Transform (ComplexType)	Specifies a BizTalk Server map, or transform, which is an item that represents the mapping between a source schema and destination schema.	Not required	Default value: none

Transform (OutboundTransforms Node)

The Transform node of the OutboundTransforms node of a binding file contains specific information about a BizTalk Server map that is exported with the binding file.

Nodes in the Transform node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
FullName	Attribute	xs:string	Specifies the full name of the map.	Not required	Default value: empty
AssemblyQualifiedName	Attribute	xs:string	Specifies the assembly qualified name of the map.	Not required	Default value: empty

PartyCollection Node

The PartyCollection node of a binding file is the parent node for all of the Party nodes which contain specific information about the parties that are exported with the binding file.

Nodes in the PartyCollection node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Party (PartyCollection Node)	Record	Party (ComplexType)	Specifies information about a party that is exported with the binding file.	Not required	Default value: none

Party (PartyCollection Node)

The Party node of the PartyCollection node of a binding file contains specific information about a party that is exported with the binding file.

Nodes in the Party node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the party.	Not required	Default value: empty
Aliases (Party Node)	Record	ArrayOfPartyAlias (ComplexType)	Container node for the collection of party aliases associated with the party.	Not required	Default value: none
SendPorts (Party Node)	Record	ArrayOfSendPortRef (ComplexType)	Container node for the collection of send ports associated with the party.	Not required	Default value: none

SignatureCert (Party Node)	Record	CertificateInfo (ComplexType)	Specifies the signature certificate used to verify the signature of the documents from this party.	Required	Default value: none
CustomData	Element	xs:string	A string containing custom data associated with the party.	Required	Default value: empty

Aliases (Party Node)

The Aliases node of the Party node of a binding file is the parent node for all of the PartyAlias nodes which contain specific information about a party alias that is exported with the binding file.

Nodes in the Aliases node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
PartyAlias (Aliases Node)	Record	PartyAlias (ComplexType)	Specifies an alias for a party.	Not required	Default value: none

PartyAlias (Aliases Node)

The PartyAlias node of the Aliases node of a binding file contains specific information about a party alias that is exported with the binding file.

Nodes in the PartyAlias node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the party alias.	Not required	Default value: empty
Qualifier	Attribute	xs:string	Specifies the qualifier for the	Not required	Default value:

			party alias. The qualifier is used to create a unique key for the alias		empty
Value	Attribute	xs:string	Specifies the value for the party alias. The qualifier and value pair for a party alias must be unique among party aliases.	Not required	Default value: empty
IsAutoCreated	Attribute	xs:boolean	Specifies whether the alias was automatically generated.	Required	Default value: none Set to **true** if the alias was automatically generated, otherwise set to **false**.

SendPorts (Party Node)

The SendPorts node of the Party node of a binding file is the parent node of the SendPortRef nodes which contain information about a send port referenced by the party that is exported with the binding file.

Nodes in the SendPorts node

The following table lists the properties that can be set for this node of a binding file:

Name	**Node Type**	**Data Type**	**Description**	**Restrictions**	**Comments**
SendPortRef	Record	SendPortRef (ComplexType)	Specifies information about a send port that is referenced by the party.	Not required	Default value: none

SendPortRef

The SendPortRef node of the SendPorts node of a binding file specifies the name of a send port referenced by a party that is exported with the binding file.

Nodes in the SendPortRef node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
Name	Attribute	xs:string	Specifies the name of the send port referenced by the party.	Not required	Default value: empty

SignatureCert (Party Node)

The SignatureCert node of the Party node of a binding file contains the signature certificate used to verify the signature of the documents from a party that is exported with the binding file.

Nodes in the SignatureCert node

The following table lists the properties that can be set for this node of a binding file:

Name	Node Type	Data Type	Description	Restrictions	Comments
CustomData	Element	xs:string	Specifies the signature certificate for this party.	Required	Default value: empty

Configuration Properties for Integrated BizTalk Adapters

The BizTalk Explorer object model exposes the **IReceiveLocation.CustomData** and **ISendPort.CustomData** properties that contain the adapter configuration property bag in the form of a name/value pair XML string. This name/value pair XML string is stored in a <CustomProps> element within a <TransportTypeData> element in a binding file. Most of the information in the <CustomProps> element corresponds to information that can be set for an adapter in the BizTalk Server user interface (such as the BizTalk Administration Console or BizTalk Explorer). If these values are present in a binding file then they are applied to the adapter configuration for the specified receive locations and send ports when the binding file is imported. Configuration information for all adapters is stored in the Single Sign-On database.

This section describes the configuration properties that can be set for each integrated BizTalk adapter.

The configuration data for adapters built using the Adapter Framework is stored in an <AdapterConfig> element. Since the <AdapterConfig> element specifies the VT_BSTR (vt="8") data type, the **< >** characters contained in this element must be escaped or an error will occur when you attempt to import the binding file. This causes the text for the configuration data to be less human readable than if these characters were not escaped. The following example illustrates the effect of escaping these characters from sample configuration data for a send port bound to the POP3 adapter.

TransportTypeData configuration data that does not escape the <> characters used in the <AdapterConfig> element

This configuration data is invalid because the <AdapterConfig> element specifies the VT_BSTR (vt="8") data type and the < > characters contained in the <AdapterConfig> element are not escaped:

TransportTypeData configuration data that does escape the <> characters used in the <AdapterConfig> element

Since the <AdapterConfig> element specifies the VT_BSTR (vt="8") data type, the < > characters must be escaped from the <AdapterConfig> element as seen below:

The integrated adapters that were created with the Adapter Framework include the following:

- Base EDI Adapter
- FTP Adapter
- MQSeries Adapter
- MSMQ Adapter
- POP3 Adapter
- SQL Adapter
- Windows Sharepoint Services Adapter

To view a sample string used as the TransportTypeData configuration data for each integrated adapter, please see the configuration properties topic that is associated with the adapter in this section.

In This Section

Configuration Property Variable Types

Base EDI Adapter Configuration Properties

BizTalk Message Queueing (MSMQT) Adapter Configuration Properties

File Adapter Configuration Properties

FTP Adapter Configuration Properties

HTTP Adapter Configuration Properties

MQSeries Adapter Configuration Properties

MSMQ Adapter Configuration Properties

POP3 Adapter Configuration Properties

SMTP Adapter Configuration Properties

SOAP Adapter Configuration Properties

SQL Adapter Configuration Properties

Windows Sharepoint Services Adapter Configuration Properties

Managing Artifacts

This section describes how to manage artifacts, including how to add and remove them from a BizTalk application and how to configure their properties.

Artifacts are the items contained in a BizTalk application that are required for it to run. For background information about how artifacts are used in a BizTalk application, see What Is a BizTalk Application?. For background information about artifacts, see Runtime Architecture. For information about how you manipulate artifacts when you create and modify a BizTalk application, see Creating and Modifying BizTalk Applications.

In This Section

- Managing Orchestrations
- Managing Role Links
- Managing Send Ports and Send Port Groups
- Managing Receive Ports
- Managing Receive Locations
- Managing Policies
- Managing Schemas
- Managing Maps

- Managing Pipelines
- Managing Resources
- Managing BAS Artifacts

Managing Orchestrations

This section provides instructions on using the BizTalk Server Administration console to manage orchestrations. An orchestration is an executable business process. For background information about orchestrations, see Orchestrations.

Orchestrations are built in Visual Studio 2005 and compiled into BizTalk assemblies. You cannot add an orchestration to an application individually; an orchestration is added to an application as follows:

- When you add a BizTalk assembly containing an orchestration to the application, as described in How to Add a BizTalk Assembly to an Application.
- When you import an .msi file into an application that includes a BizTalk assembly containing an orchestration, as described in How to Import a BizTalk Application.
- When a developer deploys into an application an assembly containing an orchestration from Visual Studio, as described in Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

You use the administration console to perform the following actions, as described in this section:

- Configure bindings for the orchestration by binding the orchestration to the appropriate send and receive ports and host, or remove these bindings from the orchestration.
- Configure message tracking for the orchestration.
- View instance information for the orchestration.
- Enlist the orchestration to the appropriate host or unenlist the orchestration from the host.
- Start or stop the orchestration so that it starts or stops processing messages.

In This Section

- How to Configure Bindings for an Orchestration
- How to Unbind an Orchestration
- How to Configure Tracking for an Orchestration
- How to View Instance Information for an Orchestration
- How to Remove an Orchestration from an Application
- How to Enlist an Orchestration
- How to Unenlist an Orchestration
- How to Start an Orchestration
- How to Stop an Orchestration
- How to Suspend, Resume, and Terminate Orchestration Instances

How to Configure Bindings for an Orchestration

This topic describes how to use the BizTalk Server Administration console to configure bindings for an orchestration. This involves binding the logical ports defined for the orchestration to physical ports in the staging or production environment as well as binding the orchestration to a host. If the orchestration has already been bound, you can change the bindings by using this procedure.

After configuring bindings, you can enlist the orchestration and then start it so that it begins processing messages. For instructions on performing these tasks, see How to Enlist an Orchestration and How to Start an Orchestration.

For background information on bindings, see Deployment and Binding.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure bindings for an orchestration

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, expand Applications, and then expand the application containing the orchestration for which you want to configure bindings

3. Click **Orchestrations**, right-click the orchestration for which you want to configure bindings, and then click **Properties**.

4. Click the **Bindings** tab, and from the **Hosts** list, select the host on which to enlist an orchestration.

5. From the drop-down lists in the **Receive Ports** column, next to each inbound logical port, select the receive port to which you want to bind the logical port.

6. From the drop-down list in the **Send Ports/Send Port Groups** column, next to each inbound logical port, select the send port to which you want to bind the logical port, and then click **OK**.

How to Unbind an Orchestration

This topic describes how to use the BizTalk Server Administration console to remove receive port, send port, or host bindings from an orchestration.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions,

To remove bindings from an orchestration

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, expand Applications, and then expand the application containing the orchestration from which you want to remove bindings

3. Click **Orchestrations**, right-click the orchestration, and click **Properties**, and then click **Bindings** in the left pane.

4. To remove the host bindings, from the **Hosts** list, select **<None>**.

5. To remove receive port bindings, from the drop-down list under **Receive Ports**, click **<None>**.

6. To remove send port bindings, from the drop-down list under **Send Ports/Send Port Groups**, click **<None>**.

7. When finished removing bindings, click **OK.**

How to Configure Tracking for an Orchestration

This topic describes how to use the BizTalk Server Administration console to configure tracking for an orchestration.

For more information about creating and using queries, see Using the BizTalk Server Administration Console. For more information about the health and activity tracking features of BizTalk Server 2006, see Health and Activity Tracking.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure tracking for an orchestration

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, expand Applications, and then expand the application containing the orchestration for which you want to configure tracking.

3. Click **Orchestrations**, right-click the orchestration for which you want to configure tracking, and then click **Properties**.

4. Click the **Tracking** tab, select the tracking options you want, as described in the following table, and then click **OK**.

Option	Description
Track Events - Orchestration start and end	Select this check box to track the orchestration instance before and after processing of the entire business process. Orchestration tracking enables you to see the instances in the reporting views of Health and Activity Tracking.
Track Events - Message send and receive	Select this check box to track message send and receive events. This check box is available only if you select the **Orchestration start and end** check box.
Track Events - Shape start and end	Select this check box when you need to debug orchestration instances in the Orchestration Debugger. When this check box is selected, the event list in the Orchestration Debugger is populated. This check box is available only if you select the **Orchestration start and end** check box.
Track Message Bodies - Messages before	Select this check box to save and track the actual message content prior to processing by the orchestration instance.

orchestration processing	This check box is available only if you select the **Message send and receive** check box.
Track Message Bodies - Messages after orchestration processing	Select this check box to save and track the actual message content after processing by the orchestration instance. This check box is available only if you select the **Message send and receive** check box.
Track Message Properties - Incoming messages	Select this check box to track message receive events. You must select this option to track incoming message bodies.
Track Message Properties - Outgoing messages	Select this check box to track message send events. You must select this option to track outgoing message bodies.

How to View Instance Information for an Orchestration

This topic describes how to view instance information for an orchestration by using the BizTalk Server Administration console. A service instance is an instance of an orchestration that BizTalk Server is either processing or has serialized into the MessageBox for further processing or tracking.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view instance information for an orchestration

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, expand Applications, and then expand the application containing the orchestration for which you want to view instance information.

3. Click **Orchestrations**, right-click the orchestration for which you want to view instance information, point to **View**, and then click **Instance Information**.

   The query results panel in the lower section of the page displays all instances of the orchestration.

   To refine the query and view different instance information for the orchestration, click the box under **Value** for the Search For field, select the instance type to view, and then click **Run Query**. For more information about creating queries, see the topics on searching under See Also.

How to Remove an Orchestration from an Application

This topic describes how to use the BizTalk Server Administration console or the command line to remove an orchestration from a BizTalk application. Removing an orchestration from an application also deletes it from the BizTalk Management database for the BizTalk group.

When you remove an orchestration, the following occurs:

- The orchestration is deleted from the BizTalk Management database.
- The BizTalk assembly that contains the orchestration is deleted from the BizTalk Management database, but is not removed from the local file system or the global assembly cache (GAC), if it exists in these locations.
- As a consequence of the BizTalk assembly being deleted, all artifacts contained in the assembly are removed deleted the BizTalk Management database as well.

Before removing an orchestration from an application, bear in mind the following important points:

- If other artifacts have dependencies on this orchestration or the artifacts contained in the assembly that will also be removed, they will no longer function correctly when you remove the orchestration. For background information about dependencies, see Dependencies and Application Deployment.
- You cannot remove an orchestration that has running instances. You must terminate any running instances.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove an orchestration from an application

Using the BizTalk Server Administration console

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, expand Applications, and then expand the application containing the orchestration that you want to remove.
3. Click **Orchestrations**, right-click the orchestration, and then click **Unenlist**.

4. Right-click the orchestration, point to **View**, and then click **Instance Information**.

5. In the query results pane, right-click the orchestration instances, and then click **Terminate**.

6. Click **Orchestrations**, right-click the orchestration and click **Remove**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource [/ApplicationName:*value*] /Luid:*value* [/Server:*value*] [/Database:*value*]

   Example:

   BTSTask RemoveResource /ApplicationName:MyApplication /Luid:"MyApp.Orchestrations, Version=1.0.0.0, Culture=neutral, PublicKeyToken=0123456789ABCDEF"

Parameter	**Description**
/ApplicationName	Name of the BizTalk application containing the orchestration to delete. If the name includes spaces, you must enclose it in double quotation marks ("). If this parameter is not specified, the default application is used.
/Luid	Locally unique identifier (LUID) of the orchestration. You can obtain the LUID by using the ListApp Command.
/Server	Name of the SQL Server instance hosting the BizTalk Management database. Required if you specify the Database parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.
/Database	Name of the BizTalk Management database. Required if you specify the Server parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.

How to Enlist an Orchestration

This topic describes how to use the BizTalk Server Administration console to enlist an orchestration into a host. The orchestration must be enlisted before you can start it.

When enlisting an orchestration, bear in mind the following points:

- **The orchestration must be bound before you can enlist it.** For instructions on configuring bindings for orchestrations, see How to Configure Bindings for an Orchestration.
- **Subscriptions are automatically created.** The orchestration enlistment process creates the necessary subscriptions in the MessageBox database.
- **You must install the application.** You must install the application containing the orchestration on all of computers where the orchestration will run. For more information, see How to Install a BizTalk Application.
- **A calling orchestration must be bound to the same host as the called orchestration.** Any orchestration that is called by another orchestration must be bound to the same host as the calling orchestration.
- **You should also enlist dependent orchestrations.** If you enlist an orchestration but do not enlist any dependent orchestrations, the dependent orchestrations will not have any subscriptions. A dependent orchestration without subscriptions may drop or suspend messages because there is no subscription for the messages to match.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To enlist an orchestration

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, expand Applications, and then expand the application containing the orchestration that you want to enlist.
3. Click **Orchestrations**, right-click the orchestration to enlist, and then click **Enlist**.

The orchestration is enlisted and the appropriate subscriptions are created. The orchestration is in the stopped state. To start processing incoming messages, you must explicitly start the orchestration by right-clicking it and clicking Start. For more information, see How to Start an Orchestration.

How to Unenlist an Orchestration

This topic describes how to unenlist an orchestration by using the BizTalk Server Administration console. Unenlisting an orchestration removes it from the host. This removes the subscription so that the orchestration no longer processes messages. You must unenlist an orchestration before you can edit its bindings.

Before you can unenlist an orchestration, you must terminate any running instances, as described in How to Suspend, Resume, and Terminate Orchestration Instances.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To unenlist an orchestration

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, expand Applications, and then expand the application containing the orchestration that you want to unenlist.
3. Click **Orchestrations**, right-click the orchestration to unenlist, and then click **Unenlist**.

How to Start an Orchestration

This topic describes how to use the BizTalk Server Administration console to start an orchestration. When you start an orchestration, it begins processing incoming messages.

When starting and orchestration, bear in mind the following important points:

- Before you can start the orchestration, it must be enlisted, along with all of the send ports and send port groups associated with it. For instructions, see How to Enlist an Orchestration and How to Enlist a Send Port or Send Port Group.

- Starting an orchestration does not automatically start any associated send ports. You must start them separately, as described in How to Start a Send Port or Send Port Group.

- If you enlist the send ports and/or send port groups associated with this orchestration, but do not start them, BizTalk Server places any messages sent to this send port or send port group in the suspended queue of the host where you enlisted the send port or send port group.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To start an orchestration

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, expand Applications, and then expand the application containing the orchestration that you want to start.

3. Click **Orchestrations**, right-click the orchestration, and then click **Start**.

How to Stop an Orchestration

This topic describes how to use the BizTalk Server Administration console to stop an orchestration. Stopping an orchestration deactivates and suspends all of the arriving activation messages.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions,

To stop an orchestration

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, expand Applications, and then expand the application containing the orchestration that you want to stop.

3. Click **Orchestrations**, right-click the orchestration, and then click **Stop**.

How to Suspend, Resume, and Terminate Orchestration Instances

This topic describes how to suspend, resume, and terminate one or more running service instances of an orchestration by using the BizTalk Server Administration console. A service instance is an instance of an orchestration that BizTalk Server is either processing or has been serialized into the MessageBox for further processing or tracking.

The following describes the effects of these three operations:

- **Suspend.** This pauses the service instance. In-process messages run to completion. Messages are still routed to service instances, but are not processed.
- **Resume.** This resumes processing of suspended instances.
- **Terminate.** This terminates all message processing. The service instance is deleted from the BizTalk databases. Messages that the service instance is processing are also deleted, except for any messages that are also referenced by a service instance that is not being terminated.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view start, stop or terminate an instance of an orchestration

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, expand Applications, and then expand the application containing the orchestration for which you want to view instance information.
3. Click **Orchestrations**, right-click the orchestration for which you want start, stop, or terminate an instance, point to **View**, and then click **Instance Information**.

   The query results panel in the lower section of the page displays all instances of the orchestration.

4. To refine the query and view different instances for the orchestration, click the box under **Value** for the **Search For** field, select the instance type to view, and then click **Run Query**. For more information about creating queries, see the topics on searching under See Also.

5. Right-click the instance you want and click **Suspend**, **Resume**, or **Terminate**.

Managing Role Links

This section provides instructions on using the BizTalk Server Administration console to manage role links in a BizTalk application. A role link defines the relationship between parties involved in a business transaction and specifies the message and port types used in the interaction in both directions. For background information on role links, see Using Role Links .

Role links are built in Visual Studio 2005 and compiled into BizTalk assemblies. You cannot add a role links to an application individually; role link is added to an application as follows:

- When you add a BizTalk assembly containing a role link to the application, as described in How to Add a BizTalk Assembly to an Application.
- When you import an .msi file into an application that includes a BizTalk assembly containing a role link, as described in How to Import a BizTalk Application.
- When a developer deploys into an application an assembly containing a role link from Visual Studio, as described in Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

The BizTalk application developer creates role links to implement communications between two or more parties involved in a business transaction, such as your organization and a supplier. The developer does not necessarily specify the parties that are involved in the transaction, only their roles. To deploy an application that includes a role link to a production environment and enable actual communication between the parties, the following configuration must be performed, as described in this section:

- If it wasn't done during the development process, a party must be assigned to each role defined in the role link. This is called "enlisting" a party for a role. You can later unenlist the party if you no longer want the party to have the assigned role.
- The logical ports for each party defined within the role link must be bound to physical ports on the BizTalk host instances that will host the application.

For more information about developing role links, see Using Role Links .

In This Section

- How to Enlist or Unenlist a Party for a Role

How to Enlist or Unenlist a Party for a Role

This topic describes how to use the BizTalk Server Administration console to enlist or unenlist a party for a role. Enlisting a party for a role assigns the party to the role and unenlisting the party removes the party from the role.

When enlisting or unenlisting a party for a role, bear in mind the following points:

- You must create a party before you can enlist it. For instructions, see How to Create a Party.
- You can enlist or unenlist a party for a role without needing to restart the application.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To enlist or unenlist a party for a role

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, expand Applications, and then expand the application containing the role link for which you want to enlist or unenlist a party.
3. Click **Role Links**, and right-click the role link for which you want to enlist or unenlist a party, and then click **Properties**.
4. To enlist a party, do the following:
   - Click **Enlist** and click the party to enlist.
   - Click **Bind**, in the Send Port drop-down list, click the send to use for this party, and then click **OK** twice.
5. To unenlist a party, click the party, click **Unenlist**, and then click **OK**.

Managing Send Ports and Send Port Groups

This section provides instructions on using the BizTalk Server Administration console to create, configure, and manage send ports and send port groups in a BizTalk application. A send port specifies the location to which messages are sent and optionally responses are received. Any time a message is sent to a send port, a new instance of the send port service is created, which is called a "service instance."

A send port group is a logical grouping of send ports. When a message is sent to a send port group, it is routed to all of the associated send ports. For background information about send ports and send port groups, see Send Ports.

In This Section

- Creating and Configuring Send Ports
- Creating and Configuring Send Port Groups
- How to Enlist a Send Port or Send Port Group
- How to Unenlist a Send Port or Send Port Group
- How to Start a Send Port or Send Port Group
- How to Stop a Send Port or Send Port Group

Creating and Configuring Send Ports

This section provides instructions on using the BizTalk Server Administration console to create and configure send ports for a BizTalk application. A send port is a location to which messages are sent or from which messages are received and is uniquely identified by its name.

In This Section

- How to Create a Send Port
- How to View Instance Information for a Send Port
- How to Configure Per-instance Pipeline Properties for a Send Port
- How to Add a Send Port to a Send Port Group
- How to Configure Transport Advanced Options for a Send Port
- How to Configure Backup Transport Options for a Send Port
- How to Configure Inbound Maps for a Send Port
- How to Configure Outbound Maps for a Send Port
- How to Configure Filters for a Send Port

- How to Assign a Certificate to a Send Port
- How to Remove a Certificate from a Send Port
- How to Configure Tracking for a Send Port
- How to Delete a Send Port

How to Create a Send Port

This topic describes how to use the BizTalk Server Administration console to create a new send port. When creating a send port, you must select the type of send port to create, as follows:

- Static one-way — a preconfigured send-only port.
- Static solicit-response — a preconfigured send port that waits for a reply from the destination.
- Dynamic one-way — a send-only port that can be bound to a protocol and location at runtime based on message properties.
- Dynamic solicit-response — a send port that waits for a reply and can be bound to a protocol and location at runtime based on message properties.

After you create a send port, you can perform the following additional steps to complete the configuration:

- Configure advanced transport options, such as the number of times to retry sending messages on message failure and the service window schedule for the port, as described in How to Configure Transport Advanced Options for a Send Port.
- Configure a backup transport, in the event the primary transport fails to function, as described in How to Configure Backup Transport Options for a Send Port.
- Configure filters to determine which messages are routed to this send port from the message box, as described in How to Configure Filters for a Send Port.
- Assign a security certificate to the send port to encrypt or sign documents handled by the send port, as described in How to Assign a Certificate to a Send Port.
- Configure tracking options for messages handled by the send port, as described in How to Configure Tracking for a Send Port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application. In addition, you need to have appropriate permissions on the SSO database.

To create a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to create a send port.

3. Right-click **Send Ports**, point to **New**, and then click the type of port to create.

4. In the **Send Ports Properties** window, do the following:

Use this	**To do this**
Name	Type the name of the new send port. This name must be unique in the BizTalk group.
Transport Type	From the drop-down list, select the appropriate transport type, or transport protocol. If the port is a solicit-response port, only transport types that support solicit-response are available in the list. This property is visible only for static ports.
Configure	After you select the transport type, click **Configure** to display the **Transport Properties** dialog box, which provides transport-specific configuration options. This property is visible only for static ports. Click **Help** in the dialog box for configuration instructions.
Send handler	From the drop-down list, select the host instance on which the send adapter is running. This property is visible only for static ports.
Send pipeline	From the drop-down list, select the pipeline that processes the messages sent through this port. After you select the pipeline, you can click the adjacent ellipsis (...) button to display the **Configure Pipeline** dialog box, where you configure per-instance pipeline properties for this specific port. Click **Help** in the dialog box for configuration instructions.
Receive pipeline	From the drop-down list, select the pipeline that processes messages received through this port. Responses to messages received through this pipeline will also be sent through this pipeline. After you select the pipeline, you can click the adjacent ellipsis (...) button to display the **Configure Pipeline** dialog box, where you configure per-instance pipeline properties for this specific port. This property is visible only for Solicit-Response ports.

5. If you are creating a solicit-response send port, in the left-hand pane, click **Inbound Maps** and do the following, repeating as necessary if you want to add multiple maps:

Use this	To do this
Source Document	From the drop-down list, select the source document for the inbound map. A send port may have more than one map, but each map should have a unique source document.
Map	From the drop-down list, select the map to associate with the source and target documents.
Target Document	From the drop-down list, select the target document for the inbound map.

6. In the left-hand pane, click **Outbound Maps** and do the following, repeating as necessary if you want to add multiple maps:

Use this	To do this
Source Document	From the drop-down list, select the source document for the outbound map.
Map	From the drop-down list, select the map to associate with the source and target documents.
Target Document	From the drop-down list, select the target document for the outbound map.

7. When finished configuring the send port, click **OK**.

How to View Instance Information for a Send Port

This topic describes how to use the BizTalk Server Administration console to view a list of the running service instances of a send port. A service instance is an instance of the send port service that is created when a message is sent to the send port. When you follow the procedure in this topic, instance information displays in the Group Overview page for the send port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group or BizTalk Server Operators group. For more detailed information on permissions,

To view instance information for a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to view service instance information for a send port.

3. Click **Send Ports**, right-click the send port, point to **View**, and then click **Instance Information**.

   The query results panel in the lower section of the page displays all running instances of the send port.

4. To refine the query and view different service instance information for the send port, click the box under **Value** for the Search For field, select the instance type to view, and then click **Run Query**. For more information about creating queries, see the topics on searching under See Also.

How to Configure Per-instance Pipeline Properties for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure pipeline properties for a send port after the pipeline has been deployed into a BizTalk group. Changes that you make overwrite the default pipeline properties for this send port only, so if you want, you can configure different pipeline properties for each send port in the BizTalk group.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure per-instance pipeline properties for a send port

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 containing the send port for which you want to configure pipeline properties, expand Applications, and then expand the application containing the send port.

3. Click the **Send Ports** folder, right-click the send port, and then click **Properties**.

4. Click the ellipsis (...) to the right of the **Send Pipeline** box.

5. Configure the properties you want, and then click **OK**. Click Help on the properties page for more information.

6. If this is a solicit-response send port, click the ellipsis (...) to the right of the **Receive Pipeline** box.

7. Configure the properties you want, and then click **OK** twice. Click Help on the properties page for more information.

How to Add a Send Port to a Send Port Group

This topic describes how to use the BizTalk Server Administration console to add one or more send ports to a send port group. You can only add one-way static send ports to a send port group.

You can add a send port that exists in a different application. If you do this, you must add a reference from the application containing the send port group to the application containing the send port. For instructions, see How to Add a Reference to Another Application.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a send port to a send port group

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

2. In the console tree, expand the BizTalk group and the BizTalk application in which you want to add a send port to a send port group.

3. Click **Send Port Groups**, right-click the send port group, and then click **Properties**.

4. In **Send ports**, click the drop-down list under **Name**, and click the send port to add to the send port group. Repeat for each send port you want to add to the group. To create a new send port and add it, click **<New send port…>** and then follow the instructions in How to Create a Send Port.

5. When finished adding send ports to the send port group, click **OK**.

How to Remove a Send Port from a Send Port Group

This topic describes how to use the BizTalk Server Administration console to remove a send port from a send port group. When you do this, the send port is not deleted from the application or the BizTalk Management database.

Before you can remove a send port, it must be in a stopped or unenlisted state.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a send port from a send port group

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand the BizTalk group and the BizTalk application in which you want to remove a send port from a send port group.
3. Click **Send Port Groups**, right-click the send port group, and then click **Properties**.
4. Under **Name**, click the send port to remove, and then click **Remove**.

How to Configure Transport Advanced Options for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure transport advanced options for a send port. These options determine how messages are handled by the send port, such as the number of times to retry sending messages on message failure and the service window schedule for the port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To specify transport options for a send port

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

2. In the console tree, expand the BizTalk group and then expand the BizTalk application for which you want to configure transport advanced options for a send port.

3. Click **Send Ports**, right-click the send port to configure, and then click **Properties**.

4. In the left pane, click **Transport Advanced Options**.

5. Configure the transport options as described in the following table, and then click **OK**.

Use this	**To do this**
Retry count	Specify the number of times for the send port to resend a message on message failure. The default is 3; the allowed range is 0 to 1000.
Retry interval	Specify the interval in minutes between message resend attempts. The default is 5; the allowed range is 0 to 525600.
Priority	Set the priority of the resend attempt.
Ordered delivery	Select this check box to send messages in order of receipt.
Stop sending subsequent messages on current message failure	Select this check box to stop sending subsequent messages that follow a failed message. This option is available only when the **Ordered delivery** option is selected.
Enable routing for failed messages	Select this option to enable routing for failed messages.
Enable service window	Select this option to specify the time period each day during which the send port will be operational by specifying a start time and stop time.
Start time	Specify the time each day at which the send port starts sending messages. This option is available only when the **Enable service window** option is selected.
Stop time	Specify the time each day at which the send port stops sending messages. This option is available only when the **Enable service window** option is selected.

How to Configure Backup Transport Options for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure backup transport options for a send port. The backup transport that you specify takes effect in the event the primary transport fails to function. Configuring the primary transport is described in How to Create a Send Port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To specify transport options for a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure backup transport options for a send port.

3. Expand **Send Ports**, right-click the send port to configure, click **Properties**, and then click **Backup Transport**.

4. Configure backup transport properties as described in the following table, and then click **OK**.

Use this	To do this
Type	From the drop-down list, select the appropriate backup transport type, or transport protocol. If the port is a solicit-response port, only transport types that support solicit-response are available in the list.
Configure	After you select the backup transport type, click **Configure**, and then configure transport properties. For more information about configuring the properties, click **Help**.
Send handler	From the drop-down list, select the host instance on which the send adapter is running.
Retry count	Specify the number of times for the send port to resend a message on message failure. The default is 3; the allowed range is 0 to 1000.
Retry interval	Specify the interval in minutes between message resend attempts. The default is 5; the allowed range is 0 to 525600.

How to Configure Inbound Maps for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure inbound maps for a send port. Inbound maps are used only with dynamic or static solicit-response send ports. You use a map to apply an XSL transformation to a response message received by the port without processing the message through an orchestration. You can add an inbound map, remove a map, or change an existing map to a different one. You can add more than one map to a send port, but each map must have a unique source schema. For background information about maps, see Maps.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To edit inbound maps for a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to edit an inbound map for a send port.

3. Expand **Send Ports**, right-click the send port, click **Properties**, and then click **Inbound Maps**.

4. Configure the inbound maps as described in the following table, and then click **OK**. Repeat as needed to add or remove multiple maps.

Use this	To do this
Remove	Click to remove the selected map.
Source Document	From the drop-down list, select the source document for the inbound map.
Map	From the drop-down list, select the map to associate with the source and target documents.
Target Document	From the drop-down list, select the target document for the inbound map.

How to Configure Outbound Maps for a Send Port

This topic describes how to configure outbound maps for a send port by using the BizTalk Server Administration console. Outbound maps are used only with request-response receive ports. You use a map to apply an XSL transformation to a message sent by the send port without processing the message through an orchestration. You can add an outbound map, remove a map, or change an existing map to a different one. You can add more than one map to a send port, but each map must have a unique source schema. For background information about maps, see Maps.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To edit outbound maps for a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to edit the outbound maps for a send port.

3. Expand **Send Ports**, right-click the send port, click **Properties**, and then click **Outbound Maps**.

4. Configure outbound maps as described in the following table, and then click **OK**. Repeat as needed to add or remove multiple maps.

Use this	To do this
Remove	Click to remove the selected map.
Outbound maps - Source Document	From the drop-down list, select the source document for the outbound map.
Outbound maps - Map	From the drop-down list, select the map to associate with the source and target documents.
Outbound maps - Target Document	From the drop-down list, select the target document for the outbound map

How to Configure Filters for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure filters for a send port. You can use filters to create simple messaging or content-based routing (CBR) applications. A filter sets conditions for message properties or fields that determine which messages are routed to the send port. A filter does not filter the messages that an orchestration routes to the send port.

You can create one or more filter expressions, which consist of a message property, an operator, and a value that is validated against the property by using the operator.

For example, you might create an expression like the following:

MSMQ.MsgID = 1

With this filter, the send port group would only subscribe to messages having an MSMQ message ID of 1.

You can create additional expressions and specify that they have an AND or OR relationship with other expressions, for example:

MSMQ.MsgID = 1 OR

SMTP.From = MyServer

In this case, the send port group would subscribe to all messages that have either an MSMQ message ID of 1 or that have been sent from the SMTP server named MyServer.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure filters for a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure send port filters.

3. Expand **Send Ports**, right-click the send port, click **Properties**, and then click **Filters**.

4. Configure filters as described in the following table, and then click **OK**.

Use this	To do this
Delete	Click to delete the selected filter expression.
Move Up	Click to move the selected property ahead in the filter expression sequence.
Move Down	Click to move the selected property down in the filter expression sequence.
Property	In the list, click a message property to use in this filter expression.
Operator	Type or select the operator for the expression.
Value	Type the value to validate against the property. The accepted value type varies according to the type of property. To see what type of value is accepted for a property, hover your mouse over the property. Acceptable values are as follows: Int: (Integer) This must be a whole number. String: A character string. dateTime: A date and/or time in .NET-supported format. For more information about .NET-supported time formats, see "DateTimeFormatInfo Class" in .NET Frameworks Help.
Group by	Select **And** or **Or** to indicate the relationship between this and other filter expressions.

How to Assign a Certificate to a Send Port

This topic describes how to use the BizTalk Server Administration console to assign a security certificate to a send port. You can perform this procedure on a two-way receive location only. The certificate must exist in the Other People certificate store on the computer running BizTalk Server, or messages associated with this send port will not be processed, and errors will be logged.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To assign a certificate to a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to assign a certificate to a send port.

3. Expand **Send Ports**, right-click the send port, click **Properties**, and then click **Certificate**.

4. If the certificate exists on the local computer, click **Browse**, browse to the certificate that you want to assign to this send port, and then click **OK**. Otherwise, skip this step.

1. If the certificate does not exist on the local computer, in the **Thumbprint** box, type or paste the certificate thumbprint, and then click **OK**. The certificate thumbprint has the format HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH, where H is a hexadecimal digit.

How to Remove a Certificate from a Send Port

This topic describes how to use the BizTalk Server Administration console to remove a security certificate from a send port. When you do this, the send port will no longer encrypt messages; messages will be sent in clear text. Removing a certificate from a send port does not remove it from the certificate store.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a certificate from a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to remove a certificate from a send port.

3. Expand **Send Ports**, right-click the send port, click **Properties**, and then click **Certificates**.

4. Click **Remove certificate**, and then click **OK**.

How to Configure Tracking for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure tracking for a send port. You can select options to view message bodies and promoted properties in the reporting views of Microsoft Health and Activity Tracking (HAT). This helps you monitor the health of your BizTalk implementation and identify any bottlenecks. The tracking settings that you configure apply to all of the instances of the send port.

For background information on tracking, see Configuring Tracking. For complete information about configuring and using HAT, see Health and Activity Tracking.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure tracking for a send port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure tracking for a send port.

3. Click **Send Ports**, right-click the send port, click **Properties**, and then click **Tracking**.

4. Configure the tracking options you want, as described in the following table, and then click **OK**.

Use this	To do this
Track Message Bodies - Request message before port processing	Select this check box to enable you to save and track message content before the message is received.
Track Message Bodies - Request message after port processing	Select this check box to enable you to save and track message content after the message is received.
Track Message Bodies - Response message before port processing	Select this check box to enable you to save and track message content before the message is sent. This check box is available only for solicit-response send ports.
Track Message Bodies - Response message after port processing	Select this check box to enable you to save and track message content after the message is sent. This check box is available only for solicit-response send ports.
Track Message Properties - Request message before port processing	Select this check box to track the promoted properties of an inbound message.
Track Message Properties - Request message before port processing	Select this check box if you want to track the promoted properties of an outbound message.
Track Message Properties - Response message before	Select this check box to save and track message properties before the message is sent. This check box

port processing	is available only for solicit-response send ports.
Track Message Properties - Response message after port processing	Select this check box to save and track properties after the message is sent. This check box is available only for solicit-response send ports.

How to Delete a Send Port

This topic describes how to use the BizTalk Server Administration console to delete a send port from a BizTalk application. When you do this, the send port is also deleted from the BizTalk Management database for the group.

You can delete a send port only if it meets the following conditions:

- An orchestration is not bound to the send port. If this is the case, you can remove the binding by following the instructions in How to Unbind an Orchestration.
- The send port is both stopped and unenlisted. For instructions on stopping and unenlisting a send port, see How to How to Unenlist a Send Port or Send Port Group and How to Stop a Send Port or Send Port Group.
- The send port is not referenced by a party. For instructions on removing a reference to a send port from a party, see How to View or Edit a Party.
- The send port is not included in a send port group. For instructions on removing a send port from a send port group, see How to Remove a Send Port from a Send Port Group.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To delete a send port

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, expand Applications, and then expand the application containing the send port that you want to delete
3. Click **Send Ports**, right-click the send port, and then click **Delete**.

Creating and Configuring Send Port Groups

This section provides instructions on using the BizTalk Server Administration console to add send port groups to a BizTalk application as well as configure and delete send port groups. A send port group is a named collection of send ports that can be used to send the same message to multiple destinations. You can bind send port groups to orchestration ports or use them for Content-based Routing (CBR) just as you can for a single send port. For background information on send port groups, see Send Port Groups.

In This Section

- How to Create a Send Port Group
- How to Add a Send Port to a Send Port Group
- How to Remove a Send Port from a Send Port Group
- How to Configure Filters for a Send Port Group
- How to Delete a Send Port Group

How to Enlist a Send Port or Send Port Group

This topic describes how to use the BizTalk Server Administration console to enlist a send port or send port group. Enlisting a send port or send port group associates the send port or send port group with a BizTalk host and creates the subscriptions for the send port or send port group. If a send port group does not contain a send port, enlisting the send port group does not create any subscriptions. In addition, enlisting a send port group does not change the state of any send ports that it contains.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To enlist a send port or send port 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, expand Applications, and then expand the application containing the send port or send port group that you want to enlist.
3. Click **Send Ports** or **Send Port Groups**, right-click the send port or send port group, and then click **Enlist**.

How to Unenlist a Send Port or Send Port Group

This topic describes how to use the BizTalk Server Administration console to unenlist a send port or send port group. Unenlisting a send port or send port group eliminates all subscriptions associated with that send port or send port group. You can unenlist both running and stopped send ports or send port groups. Unenlisting a send port or send port group automatically stops it.

For example, you may want to unenlist a send port or send port group to edit its bindings, or if you want to remove the send port or send port group from the BizTalk Server environment.

You cannot unenlist the last enlisted send port within a send port group that is enlisted or running. Unenlisting a send port group does not change the state of any send ports that it contains.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To unenlist a send port or send port 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, expand Applications, and then expand the application containing the send port or send port group that you want to unenlist.

3. Click **Send Ports** or **Send Port Groups**, right-click the send port or send port group, and then click **Unenlist**.

How to Start a Send Port or Send Port Group

This topic describes how to use the BizTalk Server Administration console to start a send port or send port group. You must start a send port or send port group before it can process messages. If you start an unenlisted send port or send port group, BizTalk enlists the send port or send port group before starting it. A send port group must contain at least one send port in an enlisted state before you can start the send port group. Starting and stopping a send port group does not affect the state of any send ports that it contains.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To start a send port or send port 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, expand Applications, and then expand the application containing the send port or send port group that you want to start.
3. Click **Send Ports** or **Send Port Groups**, right-click the send port or send port group, and then click **Start**.

How to Stop a Send Port or Send Port Group

This topic describes how to use the BizTalk Server Administration console to stop a send port or send port group. When you stop a send port or send port group, it no longer processes messages. BizTalk Server suspends all activation messages to a stopped send port or send port group. Stopping a send port group does not affect the state of any send ports that it contains.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To stop a send port or send port 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, expand Applications, and then expand the application containing the send port or send port group that you want to stop.
3. Click **Send Ports** or **Send Port Groups**, right-click the send port or send port, and then click **Stop**.

Managing Receive Ports

This section describes how to create, configure, and manage receive ports for a BizTalk application by using the BizTalk Server Administration console. A receive port is logical grouping of similar receive locations. For background information on receive ports, see Receive Ports.

In This Section

- How to Create a Receive Port
- How to View Instance Information for a Receive Port
- How to Configure Authentication Options for a Receive Port
- How to Enable Routing for Failed Messages for a Receive Port
- How to Add a Receive Location to a Receive Port
- How to Configure Inbound Maps for a Receive Port
- How to Configure Outbound Maps for a Receive Port
- How to Configure Tracking for a Receive Port
- How to Delete a Receive Port

How to Create a Receive Port

This topic describes how to use the BizTalk Server Administration console to create a receive port. A receive port is a logical grouping of similar receive locations through which services interact with external partners by receiving data.

You can create the following types of receive ports:

- One-way — used for applications that drop off a message and do not wait synchronously for a reply
- Request-response — used with applications that require a response to a message

In addition to the configuration described in this topic, you can also specify tracking options for the messages handled by a receive port, as described in How to Configure Tracking for a Receive Port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To create a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to create a receive port.

3. Right-click **Receive Ports**, point to **New**, and then click **One-way Receive Port** or **Request Response Receive Port**, according to the type of port you want to create.

4. In the **Receive Port Properties** window, do the following:

Use this	To do this
Name	Type the name of the port.
No authentication	Click this option to disable authentication. This is the default.
Drop messages if authentication fails	Click this option to enable authentication but to drop unauthenticated messages.
Keep messages if authentication fails	Click this option to enable authentication and keep unauthenticated messages.
Enable routing for failed messages	Select this check box to attempt to route any message that fails processing to a subscribing application (such as another receive port or orchestration schedule). Clear the check box to suspend failed messages and generate a negative acknowledgment (NACK). The default value is cleared. For more information, see How to Enable Routing for Failed Messages for a Receive Port.

5. In the left-hand pane, click **Receive Locations**, and create a new receive location for this receive port. For instructions, see How to Create a Receive Location.

6. In the left-hand pane, click **Inbound Maps**, and do the following:

Use this	To do this
Source Document	From the drop-down list, select the source schema to use with this port.
Map	From the drop-down list, select the map you want to associate with this port.
Target Document	From the drop-down list, select the destination schema to use with this port.

7. If you are creating a request-response receive port, then in the left-hand pane, click **Outbound Maps**, and do the following:

Use this	To do this
Source Document	From the drop-down list, select the source schema to use with this port.
Map	From the drop-down list, select the map you want to associate with this port.
Target Document	From the drop-down list, select the destination schema to use with this port.

8. When finished configuring the receive port, click **OK**.

How to View Instance Information for a Receive Port

This topic describes how to use the BizTalk Server Administration console to view the service instances for a receive port. Each time the receive port receives a message, a service instance is created to process the message. When you follow the procedure in this topic, instance information displays in the Group Overview page for the receive port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view instance information for a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to view instance information for a receive port.

3. Click **Receive Ports**, right-click the receive port, point to **View**, and then click **Instance Information**.

   The query results panel in the lower section of the page displays all instances of the receive port.

4. To refine the query and view different instance information for the receive port, click the box under **Value** for the Search For field, select the instance type to view, and then click **Run Query**. For more information about creating queries, see the topics on searching under See Also.

How to Configure Authentication Options for a Receive Port

This topic describes how to use the BizTalk Server Administration console to configure message authentication options for a receive port. These options take effect when party resolution authentication is configured. The options are:

- **No authentication.** If this option is selected, the receive port will pass the message through without checking for message credentials.

- **Drop messages if authentication fails.** If this option is selected, the receive port will check message credentials using Party resolution and discard the message if authentication fails.

- **Keep messages if authentication fails.** If this option is selected, the receive port will check message credentials using Party resolution and keep the message in the suspended queue if authentication fails.

For instructions on creating and configuring a party, see How to Create a Party. For more information about party resolution authentication, see **Authenticating the Sender of a Message** and **Inbound Message Authentication**.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure authentication options for a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure authentication for a receive port.

3. Click **Receive Ports**, right-click the receive port, and then click **Properties**.

4. In the **Authentication** section, select the option you want, and then click **OK**.

How to Enable Routing for Failed Messages for a Receive Port

This topic describes how to use the BizTalk Server Administration console to enable routing for the messages processed by a receive port. When you enable this option, BizTalk Server will attempt to route any message that fails processing to a subscribing application (such as another receive port or orchestration schedule). When this option is not enabled (the default), BizTalk Server suspends failed messages and generates a negative acknowledgment (NACK). For background information about managing failed messages, see Using Failed Message Routing.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To enable routing for failed messages for a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure failed message routing for a receive port.

3. Click **Receive Ports**, right-click the receive port, and then click **Properties**.

4. Select the **Enable routing for failed messages** check box, and then click **OK**.

How to Add a Receive Location to a Receive Port

This topic describes how to use the BizTalk Server Administration console to add a new receive location to a receive port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a receive location to a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to add a receive location to a receive port.

3. Click **Receive Ports**, right-click the receive port to which you want to add a receive location, point to **New**, and then click **Receive Location**.

4. In **Name**, type a name for the new receive location.

5. Configure properties for the receive location by following the instructions in How to Create a Receive Location, and then click **OK**.

How to Configure Inbound Maps for a Receive Port

This topic describes how to use the BizTalk Server Administration console to configure inbound maps for a receive port. You use inbound maps to transform inbound messages from one schema to another; for example to transform messages received from a partner into a schema that your company uses.

You use a map to apply an XSL transformation to a message sent by the receive port without processing the message through an orchestration. You can add an inbound map, remove a map, or change an existing map to a different one. You can add more than one map to a receive port, but each map must have a unique source schema. For background information about maps, see Maps.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure inbound maps for a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure an inbound maps for a receive port.

3. Click **Receive Ports**, right-click the receive port, click **Properties**, and then click **Inbound Maps**.

4. Configure the inbound maps as described in the following table, and then click **OK**.

Use this	To do this
Remove	Click to remove the selected map.
Source Document	From the drop-down list, select the source schema to use with this port.
Map	From the drop-down list, select the map you want to associate with this port.
Target Document	From the drop-down list, select the destination schema to use with this port.

How to Configure Outbound Maps for a Receive Port

This topic describes how to use the BizTalk Server Administration console to configure outbound maps for a receive port. You can use outbound maps with request-response receive ports to transform outbound messages from one schema to another; for example to transform messages that your company uses into a schema that a partner uses.

You use a map to apply an XSL transformation to a response message sent by the receive port without processing the message through an orchestration. You can add an outbound map, remove a map, or change an existing map to a different one. You can add more than one map to a receive port, but each map must have a unique source schema. For background information about maps, see Maps.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure outbound maps for a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure outbound maps for a receive port.

3. Expand **Recieve Ports**, right-click the receive port, click **Properties**, and then click **Outbound Maps**.

4. Configure the outbound maps as described in the following table, and then click **OK**.

Use this	To do this
Remove	Click to remove the selected map.
Source Document	From the drop-down list, select the source schema to use with this port.
Map	From the drop-down list, select the map you want to associate with this port.
Target Document	From the drop-down list, select the destination schema to use with this port.

How to Configure Tracking for a Receive Port

This topic describes how to use the BizTalk Server Administration console to configure tracking for a send port. You can select options to view message bodies and promoted properties in the reporting views of Microsoft Health and Activity Tracking (HAT). This helps you monitor the health of your BizTalk implementation and identify any bottlenecks. The tracking settings that you configure apply to all of the instances of the send port.

For background information on tracking, see Configuring Tracking. For complete information about configuring and using HAT, see Health and Activity Tracking.

The tracking settings that you configure apply to all of the instances of the receive port.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure tracking for a receive port

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure tracking for a receive port.

3. Click **Receive Ports**, right-click the receive port and click **Tracking**.

4. Configure the tracking options you want, as described in the following table, and then click **OK**.

Use this	To do this
Track Message Bodies - Request message before port processing	Select this check box to save and track message content before the message is received.
Track Message Bodies - Request message after port processing	Select this check box to save and track message content after the message is received.
Track Message Bodies - Response message before port processing	Select this check box to save and track message content before the message is sent. This check box is available only for request-response receive ports.
Track Message Bodies - Response message after port processing	Select this check box to save and track message content after the message is sent. This check box is available only for request-response receive ports.
Track Message Properties - Request message before port processing	Select this check box to track the promoted properties of an inbound message.
Track Message Properties - Request message before port processing	Select this check box if you want to track the promoted properties of an outbound message.
Track Message Properties - Response message before port processing	Select this check box to save and track message properties before the message is sent. This check box is available only for request-response receive ports.
Track Message Properties - Response message after port processing	Select this check box to save and track properties after the message is sent. This check box is available only for request-response receive ports.

How to Delete a Receive Port

This topic describes how to use the BizTalk Server Administration console to delete a receive port from a BizTalk application. When you do this, the receive port is also deleted from the BizTalk Management database for the group, as are all receive locations in this receive port.

For this operation to succeed, the receive port cannot be bound to an orchestration. For instructions on removing the bindings for a receive port, see How to Unbind an Orchestration.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To delete a receive port

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, expand Applications, and then expand the application containing the receive port that you want to delete

3. Click **Receive Ports**, right-click the receive port, and then click **Delete**.

Managing Receive Locations

This section provides instructions on creating, configuring, and managing receive locations for a BizTalk application by using the BizTalk Server Administration console. A receive location consists of an address at which inbound messages arrive and the messaging pipeline that processes the message received at that address. For background information, see Receive Locations.

In This Section

- How to Create a Receive Location
- How to Edit the Properties of a Receive Location
- How to Configure Per-instance Pipeline Properties for a Receive Location
- How to Enable a Receive Location
- How to Disable a Receive Location
- How to Configure Scheduling for a Receive Location
- How to Assign a Certificate to a Receive Location
- How to Remove a Certificate from a Receive Location
- How to Delete a Receive Location

How to Create a Receive Location

This topic describes how to use the BizTalk Server Administration console to create a new receive location and specify the receive port to which it belongs. A receive location is an address where inbound messages arrive as well as the messaging pipeline that processes messages received at that address.

Before you can create a receive location, a receive port must already exist in this application that of the same type as the receive location you want to create. For instructions on creating a receive port, see How to Create a Receive Port.

You can create the following types of receive locations:

- One-way — used for applications that drop off a message and do not wait synchronously for a reply
- Request-response — used with applications that require a response to a message

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application. In addition, you need to have appropriate permissions on the SSO database.

To create a receive location

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to create a receive location.
3. Right-click **Receive Locations**, point to **New**, and then click **One-way Receive Location** or **Request Response Receive Location**, according to the type of receive location to create.
4. In the Select a Receive Port window, click the receive port that will contain this receive location, and then click **OK**.
5. In the **Receive Location Properties** window, do the following, and then click **OK**:

Use this	**To do this**
Name	Type the name of the receive location.
Type	From the drop-down list, select the transport type, or transport protocol. If you change the transport type, you must configure transport properties, as described next.
Configure	After you select the transport type, click **Configure** to display the **Transport Properties** dialog box and configure adapter properties for the receive location. For instructions, click **Help** in the dialog box. For details on configuring each type of adapter, see the appropriate topic under Using Adapters .
Receive handler	From the drop-down list, select the instance of the BizTalk Server host on which the receive location will run. The receive handler must be running on this host.
Receive pipeline	From the drop-down list, select the receive pipeline to use to receive messages at this receive location.
Send pipeline	From the drop-down list, select the send pipeline to use to send responses from this receive location. This list is available only for a receive location associated with a request-response receive port.
Make this the primary location	Select this check box if you have more than one receive location for a receive port and you want this receive location to represent the receive port when the port needs to be passed to another entity, such as a business partner, that needs to send messages to your organization. The first receive location created is automatically selected as the primary receive location. This property remains selected and unavailable until you designate a different receive location as the primary.

How to Edit the Properties of a Receive Location

This topic describes how to use the BizTalk Server Administration console to edit properties of an existing receive location. For instructions on creating a receive location, see How to Create a Receive Location.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To edit the properties of a receive location

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to edit the properties of a receive location and then click **Receive Locations**.

3. In the right pane, right-click the receive location, and then click **Properties**.

4. In the **Receive Location Properties** window, edit one or more of the following properties, and then click **OK**:

Use this	To do this
Name	Type the name of the receive location.
Type	From the drop-down list, select the transport type, or transport protocol. If you change the transport type, you must configure transport properties, as described next.
Configure	After you select the transport type, click **Configure** to display the **Transport Properties** dialog box and configure adapter properties for the receive location. For instructions, click **Help** in the dialog box. For details on configuring each type of adapter, see the appropriate topic under Using Adapters .
Receive handler	From the drop-down list, select the instance of the BizTalk Server host on which the receive location will run. The receive handler must be running on this host.
Receive pipeline	From the drop-down list, select the receive pipeline to use to receive messages at this receive location.
Send pipeline	From the drop-down list, select the send pipeline to use to send responses from this receive location. This list is available only for a receive location associated with a request-response receive port.
Make this the primary location	Select this check box if you have more than one receive location for a receive port and you want this receive location to represent the receive port when the port needs to be passed to another entity, such as a business partner, that needs to send messages to your organization. The first receive location created is automatically selected as the primary receive location. This property remains selected and unavailable until you designate a different receive location as the primary.

How to Configure Per-instance Pipeline Properties for a Receive Location

This topic describes how to use the BizTalk Server Administration console to configure pipeline properties for a receive location after the pipeline has been deployed into a BizTalk group. Changes that you make overwrite the default pipeline properties for this receive location only, so if you want, you can configure different pipeline properties for each receive location in the BizTalk group.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure per-instance pipeline properties for a receive location

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 containing the receive location for which to configure pipeline properties, expand Applications, and then expand the application containing the receive location.

3. Click the **Receive Locations** folder, right-click the receive location, and then click **Properties**.

4. Click the ellipsis (...) to the right of the **Receive Pipeline** box.

5. Configure the properties you want, and then click **OK**. For more information, click **Help** on the properties page.

6. If this is a request-response receive location, click the ellipsis (...) to the right of the **Send Pipeline** box.

7. Configure the properties you want, and then click **OK** twice. For more information, click **Help** on the properties page

How to Enable a Receive Location

This topic describes how to use the BizTalk Server Administration console to enable a receive location. A receive location must be enabled before it can receive messages.

For instructions on creating a receive location, see How to Create a Receive Location.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To enable a receive location

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to enable a receive location.

3. Click **Receive Locations**, right-click the receive location, and then click **Enable**.

How to Disable a Receive Location

This topic describes how to use the BizTalk Server Administration console to disable a receive location. When you disable a receive location, it stops receiving messages. If you want, after disabling a receive location, you can enable it again, so that it will again receive messages.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Operators group or the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To disable a receive location

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to disable a receive location.

3. Click **Receive Locations**, right-click the receive location, and then click **Disable**.

How to Configure Scheduling for a Receive Location

This topic describes how to use the BizTalk Server Administration console to configure scheduling properties for a receive location. You can specify the dates when you want the receive location to start and stop processing messages. You can also specify certain times of the day during which you want the receive location to process messages.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure scheduling for a receive location

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure scheduling for a receive location.
3. Click **Receive Locations**, right-click the receive location, and click **Properties**.
4. In the left pane, click **Schedule**, configure scheduling properties as described in the following table, and then click **OK**.

Use this	To do this
Start date	Select this check box, and then from the pull-down calendar, select the date on which the receive location starts processing messages. To change the year, click the displayed year.
Stop date	Select this check box, and then from the pull-down calendar, select the date on which the receive location stops processing messages. This property is optional. If you do not specify a stop date, the receive location remains active until it is disabled.
Enable service window	Select this check box to configure the receive location to receive messages only at specified times of the day, then specify the times in the **Start time and Stop time** boxes. If the check box is cleared, the receive location receives messages at any time. The default value is false (cleared).
Start time	Specify the time when the receive location should begin to receive messages. This box is available only when the **Enable service window** check box is selected.
Stop time	Specify the time when the receive location should cease to receive messages. This box is available only when the **Enable service window** check box is selected.

How to Assign a Certificate to a Receive Location

This topic describes how to use the BizTalk Server Administration console to assign a security certificate to a receive location. You can perform this procedure on a two-way receive location only. The certificate must exist in the Other People certificate store on the computer running BizTalk Server, or messages associated with this receive location will not be processed, and errors will be logged.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To assign a certificate to a receive location

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to assign a certificate to a receive location.

3. Expand **Receive Locations**, right-click the receive location, click **Properties**, and then click **Certificate**.

4. If the certificate exists on the local computer, click **Browse**, browse to the certificate that you want to assign to this receive location, and then click **OK**. Otherwise, skip this step.

1. If the certificate does not exist on the local computer, in the **Thumbprint** box, type or paste the certificate thumbprint, and then click **OK**. The certificate thumbprint has the format HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH, where H is a hexadecimal digit.

How to Remove a Certificate from a Receive Location

This topic describes how to use the BizTalk Server Administration console to remove a security certificate from a receive location. When you do this, the receive location will no longer encrypt messages; messages will be sent in clear text. Removing a certificate from a receive location does not remove it from the certificate store.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a certificate from a receive location

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to remove a certificate from a receive location.

3. Expand **Receive Locations**, right-click the receive location, click **Properties**, and then click **Certificates**.

4. Click **Remove certificate**, and then click **OK**.

How to Delete a Receive Location

This topic describes how to use the BizTalk Server Administration console to delete a receive location. When you delete a receive location, it is removed from the BizTalk Management database and is no longer displayed in the BizTalk Server Administration console.

When deleting a receive location, bear in mind the following important points:

- Before you can delete a receive location, it must first be disabled, as described in How to Disable a Receive Location.

- You cannot delete the primary receive location for a receive port. If you attempt this, you will receive an error message. To delete the receive location, you can either delete the receive port, which also deletes all of the receive locations that it contains, or you can make a different receive location primary and then delete the original receive location. For instructions on making a receive location the primary receive location, see How to Edit the Properties of a Receive Location.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To delete a receive location

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to delete a receive location.

3. **Click Receive** Locations, right-click the receive location to delete, and then click **Delete**.

Managing Policies

The topics in this section provide instructions on using the BizTalk Server Administration console or the BTSTask command-line tool to manage policies. A policy is a logical grouping of business rules. For background information on policies, see Policies .

Solution developers can create and view policies by using the Business Rule Composer, as described in Creating Business Rules . Developers and IT administrators can then perform the following tasks, which are described in the topics in this section, to deploy and manage policies in a BizTalk group and application:

- **Import the policy into a BizTalk group.** When you do this, the policy is added to the Rule Engine database for the group and displays in the BizTalk Server Administration console in the <All Artifacts> node for the BizTalk group. This does not put the policy into effect for any particular application. You must first publish the policy, add it to an application, and then deploy it, as described in other topics in this section. The Rule Engine database is a database that contains all of the policies in a BizTalk group.
- **Publish a policy.** This makes it available to use in a BizTalk application.
- **Add a policy to a BizTalk application.** This allows the application to use the policy, but does not put the policy into effect. The policy takes effect when it is deployed.
- **Deploy a policy.** Doing this puts it into operation. (This is similar to starting an orchestration.) You can deploy and undeploy a policy manually. In addition, when an application is started, its policies are automatically deployed, and when an application is stopped, its policies are automatically undeployed.
- **Remove a policy from a BizTalk application and the BizTalk group.** This undeploys the policy and removes it from the application as well as the Rule Engine database for the group.
- **Export the policy.** You can then import it into a different BizTalk group to use there.

In This Section

- How to Import a Policy
- How to Publish a Policy
- How to Add a Policy to an Application
- How to Deploy or Undeploy a Policy
- How to Configure Tracking for a Policy

- How to Remove a Policy from an Application and the BizTalk Group
- How to Export a Policy

How to Import a Policy

This topic describes how to use the BizTalk Server Administration console to import a policy into a BizTalk group or the BTSTask command-line tool to import a policy into a BizTalk application.

You can create a policy by using the Business Rule Composer, as described in Creating Business Rules , and then import it directly, or you can export a policy from another BizTalk group, as described in How to Export a Policy and then import it.

Importing a policy registers it in the Rule Engine database for the BizTalk group. After you import the policy, you can view it in the BizTalk Server Administration console. If you use the BizTalk Server Administration console to import a policy, it will display in the <All Artifacts> node for the BizTalk group. You can then publish it to make it available to add it to a BizTalk application, as described in How to Publish a Policy. If you use the BTSTask command-line tool to import a policy, the policy will be automatically published and will display in the Policies folder of the application into which you imported it.

When importing a policy, bear in mind the following important points:

- Even if you specify the option to overwrite an existing policy with the imported policy, you cannot import a policy that already exists in the Rule Engine database for the group and has been deployed. The import operation will fail.
- Even if the policy was in a deployed state when exported from another BizTalk group, it will be in an undeployed state when imported.
- BTSTask does not provide a specific command for importing policies; however you can use the ExportApp command of BTSTask to selectively export only the policies in an application that you want, including no other application artifacts. Then you can use the ImportApp command to import the .msi file into an application in a different BizTalk group. This is the approach described in this topic. When you do this, the policy is automatically imported and published in the BizTalk group and added to the specified application.

For more information about working with policies, see Managing Policies. For best practices on adding policies to applications, see Best Practices for Deploying a BizTalk Application.

Prerequisites

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

- You must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

- The Business Rule Engine must be installed. For more information, see Quick Start Guide to Installing and Configuring BizTalk Server 2006 .

- If you want to use the BizTalk Server Administration console to import a policy, you must have available an .xml file containing the policy that you want to import. You can generate such an .xml file by exporting a policy from another BizTalk group or application, as described in How to Export a Policy, or by using the Business Rule Composer, as described in How to Import and Export Policies and Vocabularies .

- If you want to use BTSTask to import a policy, you must have an .msi file containing the policy to import. For instructions, see How to Export a Policy.

To import a policy

Using the BizTalk Server Administration console

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 into which you want to import the policy, and then expand <All Artifacts>.

3. Right-click **Policies**, and then click **Import**.

4. Browse to the .xml file containing the policy and click **Open**.

   The policy is imported into the group and displays in the Policies folder of <All Artifacts>.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask ImportApp /Package:*value* [**/ApplicationName:***value*] [**/Overwrite**] [**/Server:***value*] [**/Database:***value*]

   Example:

   BTSTask ImportApp /Package:"C:\MSI Files\MyApplication.msi" /Environment:Test /ApplicationName:MyApplication /Overwrite

Parameter	Value
/Package	Full path of the .msi file containing the policy to import. If the path includes spaces, you must enclose it in quotation marks (").
/ApplicationName	Name of the BizTalk application into which to import the policy. If not specified, the application name that was specified when exporting the .msi file is used. If the specified application does not exist, it will be created. Application names that include spaces must be enclosed with double quotation marks (").
/Overwrite	Option to overwrite policies in the application with artifacts in the .msi file that have the same name and version number. If this option is not specified, and there are one or more policies in the application that have the same name and version number as policies in the .msi file, the import fails. You can view the name and version number of the policies in an application by using the ListApp Command.
/Server	Name of the SQL Server instance hosting the BizTalk Management database, in the form ServerName\InstanceName,Port. Instance name is only required when the instance name is different than the server name. Port is only required when SQL Server uses a port number other than the default (1433). Examples: Server=MyServer Server=MyServer\MySQLServer,1533 If not provided, the name of the SQL Server instance running on the local computer is used.
/Database	Name of the BizTalk Management database. If not specified, the BizTalk Management database running in the local instance of SQL Server is used.

How to Publish a Policy

This topic describes how to use the BizTalk Server Administration console to publish a policy in a BizTalk group. Publishing a policy makes it available to add to a BizTalk application, as described in How to Add a Policy to an Application.

Before you can publish a policy, it must exist in the Rule Engine database for the BizTalk group. There are three ways to import a policy into the Rule Engine database:

- You can import an application that contains a policy. When you do this, the policy is automatically imported into the Rule Engine database.

- You can explicitly import a policy into the Rule Engine database by using the administration console or BTSTask, as described in How to Import a Policy.

- You can add a policy to the Rule Engine database by using the Rule Engine Deployment Wizard, as described in How to Deploy and Undeploy Policies and Vocabularies .

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To publish a policy

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 containing the policy to publish, and then expand <All Artifacts>.

3. Click **Policies**, right-click the policy, and then click **Publish**.

How to Add a Policy to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a policy to a BizTalk application. When using the administration console, you can add more than one policy at a time. Adding a policy to an application makes it available for use by that application as well as any other applications that reference it.

When adding a policy to an application, bear in mind the following important points:

- Before you can add a policy to an application, the policy must exist in the Rule Engine database for the BizTalk group, and it must be published, as described in How to Import a Policy.

- The policy cannot already exist in another application in the BizTalk group.

- For the policy to take effect and begin functioning, it must also be deployed. Policies are automatically deployed when the application starts, or you can manually deploy them as described in How to Deploy or Undeploy a Policy.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a policy to an application

Using the BizTalk Server Administration console

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 and the BizTalk group.

3. Expand Applications, expand the application to which you want to add a policy, and then right-click **Policies**.

4. Point to **Add** and click **Policy**.

5. Select the check box of each policy and version to add, and then click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [/ApplicationName:*value*] /Type:System.BizTalk:Rules [Overwrite] /Name:*value* /Version:*value* [/Server:*value*] [/Database:*value*]

   Example:

   BTSTask ExportApp /ApplicationName:MyApplication /Type:System.BizTalk:Rules /Overwrite /Name:MyPolicy /Version:1.0 /Server:MyDatabaseServer /Database:BizTalkMgmtDb

Parameter	Value
/ApplicationName	Name of the BizTalk application to which to add the policy. If the application name is not specified, the default BizTalk application for the group is used. Names that include spaces must be enclosed in double quotation marks (").
/Type	**System.BizTalk:Rules**

/Overwrite	Option to update an existing policy. If not specified, and a policy already exists in the application that has the same name as the policy being added, the AddResource operation fails.
/Name	Name of the policy.
/Version	Version number of the policy.
/Server	Name of the SQL Server instance hosting the BizTalk Management database. Required if you specify the Database parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.
/Database	Name of the BizTalk Management database. Required if you specify the Server parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.

How to Deploy or Undeploy a Policy

This topic describes how to use the BizTalk Server Administration console to deploy or undeploy a policy manually. In addition, starting an application automatically deploys any policies it contains, and stopping an application automatically undeploys its policies. Deploying a policy puts it into effect in the application that uses it. Undeploying a policy makes it inactive so that it no longer functions in any application that uses it in the BizTalk group.

When deploying or undeploying a policy, bear in mind the following important points:

- Before you can deploy a policy, it must exist in the Rule Engine database for the BizTalk group. If you import an application that contains a policy, the policy is automatically imported into the Rule Engine database, or you can explicitly import a policy into the Rule Engine database by using the administration console or BTSTask, as described in How to Import a Policy. Alternatively, you can add a policy to the Rule Engine database by using the Rule Engine Deployment Wizard, as described in How to Deploy and Undeploy Policies and Vocabularies .

- Before you can deploy a policy, it must be published, as described in How to Publish a Policy.

- A deployed policy cannot be modified. If you want to modify a deployed policy, you must first undeploy it.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To deploy or undeploy a policy

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 containing the policy that you want to deploy or undeploy, and then expand <All Artifacts>.

3. Click **Policies**, right-click the policy, and then click **Deploy** or **Undeploy**.

How to Configure Tracking for a Policy

This topic describes how to use the BizTalk Server Administration console to configure tracking for a policy. You can select options to view instance data, results of conditions, actions, and agenda updates in the query views of the administration console Group Hub page.

For more information about creating and using queries, see Using the BizTalk Server Administration Console. For more information about the health and activity tracking features of BizTalk Server 2006, see Health and Activity Tracking.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure tracking for a policy

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

2. In the console tree, expand the BizTalk group and the BizTalk application for which you want to configure tracking for a policy.

3. Click **Policies**, right-click the policy, click **Properties**, and then click **Tracking**.

4. Select the tracking options you want, as described in the following table, and then click **OK**.

Use this	To do this
Fast activity	Select this check box to track the instance data on which the policy operates.
Condition evaluation	Select this check box to track the true/false results of conditions in the selected policy.
Rule firings	Select this check box to track the actions started as a result of the policy.
Agenda updates	Select this check box to track updates to the agenda. The agenda contains a list of actions that are "true" and need to fire.

How to Remove a Policy from an Application and the BizTalk Group

This topic describes how to use the BizTalk Server Administration console or the command-line to remove a policy from an application and the Rule Engine database for the BizTalk group.

Before removing a policy, bear in mind the following important points:

- You cannot remove a deployed policy. You must first undeploy the policy, as described in How to Deploy or Undeploy a Policy. Undeploying a policy makes it inactive so that it no longer functions in any application that uses it in the BizTalk group.

- Removing a policy deletes it from the Rule Engine database. If you want to use this policy again, you should export it to an .xml file before removing it. For instructions, see How to Export a Policy.

- If the policy is used by other applications, it will no longer be in effect for the other applications. Verify that you no longer want to use the policy for any other applications that reference it before you remove it.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a policy

Using the BizTalk Server Administration console

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 containing the policy to remove, and then expand the application containing the policy

3. Click **Policies**, right-click the policy, and then click **Remove.**

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK.**

2. Type the following command, substituting the appropriate values, as described in the following table:

   ```
   BTSTask RemoveResource [/ApplicationName:value] /Luid:value [/Server:value] [/Database:value]
   ```

   Example:

   ```
   BTSTask RemoveResource /ApplicationName:MyApplication /Luid:"Rule/Policy1/1.0"
   ```

Parameter	Description
/ApplicationName	Name of the BizTalk application containing the policy to delete. If this parameter is not specified, the default application is used.
/Luid	Locally unique identifier (LUID) of the policy. You can obtain the LUID by using the **ListApp** command, as described in ListApp Command.
/Server	Name of the SQL Server instance hosting the BizTalk Management database. Required if you specify the Database parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.
/Database	Name of the BizTalk Management database. Required if you specify the Server parameter. If Server and Database parameters are not specified, the default BizTalk Management database for the group is used.

How to Export a Policy

This topic describes how to use the BizTalk Server Administration console or the command line to export one or more policies and associated vocabularies.

When exporting a policy, bear in mind the following important points:

- Using the BizTalk Server Administration console, you can export policies from a BizTalk group or a BizTalk application as well as the vocabularies to export. Using BTSTask, you can export policies from an application, and all of the associated vocabularies will be exported as well. You cannot select the vocabularies to export.

- You can then import the policy or policies into a different BizTalk group or an application in a different BizTalk group, as described in How to Import a Policy.

- Before you can export a policy, it must exist in the Rule Engine database for the BizTalk group. There are several ways to import a policy into the Rule Engine database, as described in How to Import a Policy.

- When you use the administration console for exporting, the policies and vocabularies are exported into an .xml file. When you use the BTSTask command-line tool for exporting, the policies and vocabularies are exported into an application .msi file.

- BTSTask does not provide a specific command for exporting policies; however you can use the ExportApp command of BTSTask to selectively export only the policies you want, and no other artifacts. This generates an application .msi file containing the policies. You can use the ImportApp command to import the .msi file into a different BizTalk group.

Prerequisites

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

- You must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information about permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

- The Business Rule Engine must be installed. For more information, see Quick Start Guide to Installing and Configuring BizTalk Server 2006 .

- The policy that you want to export must exist in the Rule Engine database for the BizTalk group. If you want to export the policy from an application, it must have also been added to the application as well.

To export a policy

Using the BizTalk Server Administration console

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 and expand the BizTalk group.

3. If you want to select the policies to export from all of the policies in a BizTalk group right-click the **Applications** folder, click **Export**, and then click **Policies**.

   OR

   If you want to export the policies in a particular application, expand the Applications folder, right-click the application, click **Export**, and then click **Policies**.

   OR

   If you want to export only a particular policy, click the Policies folder that contains the policy, right-click the policy, and then click **Export**.

4. On the Export Policies page, in **Policies to export**, select the policies to export.

5. In **Vocabularies to export**, select the check boxes of the vocabularies to export, and clear the checkboxes of any vocabularies you do not want to export. The vocabularies used by this policy are automatically selected.

6. In **File to export** into, type the path of the XML file to which to export the policy or policies, and then click **OK**.

Using the command line

1. Use the BTSTask ListApp command with the /ResourceSpec option to generate an XML file that lists the artifacts in the BizTalk application from which you want to export a policy, as described in ListApp Command.

2. Edit the XML file generated in the previous step, deleting all of the artifacts except for the policy or policies that you want to export.

3. Use the BTSTask ExportApp command, and specify the modified XML file for the /ResourceSpec parameter. For more information, see ExportApp Command.

   BTSTask exports the specified policies and all of their associated vocabularies into an application .msi file.

Managing Schemas

This section provides instructions on using the BizTalk Server Administration console to manage schemas. Schemas are used by pipelines and orchestrations to represent the message that will be processed.

Schemas are created in Visual Studio 2005 and compiled into BizTalk assemblies. You cannot add a schema to an application individually; a schema is added to an application as follows:

- When you add a BizTalk assembly containing a schema to the application, as described in How to Add a BizTalk Assembly to an Application.
- When you import an .msi file into an application that includes a BizTalk assembly containing a schema, as described in How to Import a BizTalk Application.
- When a developer deploys into an application an assembly containing a schema from Visual Studio, as described in Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

For background information about schemas, see Schemas. For information about developing schemas, see Creating Schemas Using BizTalk Editor .

In This Section

- How to Show and Hide Property Schemas
- How to View the Schema Definition (XSD) of a Schema
- How to Configure Tracking for a Schema
- How to Remove a Schema from an Application

How to Show and Hide Property Schemas

This topic describes how to use the BizTalk Server Administration console to show and hide property schemas in the Schemas folder for an application. A property schema is a simple version of a BizTalk schema that plays a role in the process of copying promoted properties back and forth between the instance message and the message context. You might want to hide property schemas to simplify the schema view, so that you only see document schemas.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group or BizTalk Server Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To show or hide property schemas in the Schemas folder

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 containing the schemas folder in which you want to show or hide property schemas, and then expand the application containing the schemas folder.

3. Right-click the **Schemas** folder, and click either **Hide Property Schemas** or **Show Property Schemas**.

How to View the Schema Definition (XSD) of a Schema

This topic describes how to use the BizTalk Server Administration console to view the schema definition (XSD) of a schema. You might want to do this to troubleshoot schema validation errors or validate that the correct schema is deployed.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group or BizTalk Server Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view the XSD of a schema

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 containing the schema for which you want to view the XSD, and then expand the application containing the schema.

3. Click **Schemas**, right-click the schema, and then click **Properties**.

4. In the left pane, click **Schema View**.

   The XSD displays in the right pane.

How to Configure Tracking for a Schema

This topic describes how to use the BizTalk Server Administration console to configure tracking for a schema. To configure tracking, you specify the properties of the messages that you want to view in the query views of the administration console Group Hub page.

For more information about creating and using queries, see Using the BizTalk Server Administration Console. For more information about the health and activity tracking features of BizTalk Server 2006, see Health and Activity Tracking. For background information about tracking message properties, see **Message Properties**.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. To you want to view tracking options only, you can be logged on as a member of the BizTalk Server Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure tracking for a schema

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 containing the schema for which you want to configure tracking, and then expand the application containing the schema.

3. Click **Schemas**, right-click the schema, and then click **Properties**.

4. In the left pane, click **Tracking**.

5. Do one of the following to specify which properties to use for tracking messages, and then click **OK**:

   - Select the **Always track all properties** check box use all message properties regardless of the schema version. This check box is available only for document schemas.

   - Select the **Select all message properties** check box to use all the listed properties.

   - Under **Properties list**, select the check box of each property that you want to use.

How to Remove a Schema from an Application

This topic describes how to use the BizTalk Server Administration console to remove a schema from an application. This procedure removes the schema from the BizTalk Management database for the group as well. You might want to remove a schema after deploying a new version of the schema. For more information and important considerations for updating application artifacts, see Application Versioning.

When you remove a schema, and a previous version of the schema having the same root namespace exists in the application, the previous version will become active.

When you remove a schema, the following occurs:

- The schema is deleted from the BizTalk Management database.
- The BizTalk assembly that contains the schema is deleted from the BizTalk Management database, but is not removed from the local file system or the global assembly cache (GAC), if it exists in these locations.
- As a consequence of the BizTalk assembly being deleted, all artifacts contained in the assembly are removed deleted the BizTalk Management database as well.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a schema

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 containing the schema to remove and the application containing the schema.
3. Click **Schemas**, right-click the schema, and then click **Remove**.

Managing Maps

This section provides instructions on managing maps by using the BizTalk Server Administration console. BizTalk Server uses maps to translate the records and fields in one schema to the records and fields in another schema. Maps may be used by orchestrations, send ports, and receive ports.

Maps are built in Visual Studio 2005 and compiled into BizTalk assemblies. You cannot add a map to an application individually; a map is added to an application as follows:

- When you add a BizTalk assembly containing a map to the application, as described in How to Add a BizTalk Assembly to an Application.
- When you import an .msi file into an application that includes a BizTalk assembly containing a map, as described in How to Import a BizTalk Application.
- When a developer deploys into an application an assembly containing a map from Visual Studio, as described in Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

For background information about maps, see Maps. For information about creating maps, see Creating Maps Using BizTalk Mapper .

In This Section

- How to View the Maps for an Application
- How to Remove a Map from an Application

How to View the Maps for an Application

This topic describes how to use the BizTalk Server Administration console to view the maps contained in a BizTalk application.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group or BizTalk Server Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view the maps for an application

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 containing the application whose maps you want to view, and then expand the application.
3. Click the **Maps** folder.

How to Remove a Map from an Application

This topic describes how to use the BizTalk Server Administration console to remove a map from a BizTalk application. You might want to remove a map after deploying a new version of the map. For more information and important considerations for updating application artifacts, see Application Versioning.

When you remove a map, the following occurs:

- The map is deleted from the BizTalk Management database.
- The BizTalk assembly that contains the map is deleted from the BizTalk Management database, but is not removed from the local file system or the global assembly cache (GAC), if it exists in these locations.
- As a consequence of the BizTalk assembly being deleted, all artifacts contained in the assembly are removed deleted the BizTalk Management database as well.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a map

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 containing the map to remove, and then expand the application containing the map.
3. Click the **Maps** folder, right-click the map, and then click **Remove**.

Managing Pipelines

This section provides instructions on using the BizTalk Server Administration console to manage the pipelines in a BizTalk group. You can configure tracking for events and messages as well as configure pipeline properties for a specific send port or receive location.

Pipelines perform actions on incoming and outgoing messages, such as transforming them from a native format into XML, encrypting or unencrypting data, performing property promotion, and so on.

You cannot add a pipeline to an application individually; a pipeline is added to an application as follows:

- When you add a BizTalk assembly containing a pipeline to the application, as described in How to Add a BizTalk Assembly to an Application.

- When you import an .msi file into an application that includes a BizTalk assembly containing a pipeline, as described in How to Import a BizTalk Application.

- When a developer deploys into an application an assembly containing a pipeline from Visual Studio, as described in Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application.

For background information about pipelines, see Pipelines. For information about developing pipelines, see Creating Pipelines Using Pipeline Designer .

In This Section

- How to Configure Tracking for a Pipeline

- How to Configure Per-instance Pipeline Properties for a Send Port

- How to Configure Per-instance Pipeline Properties for a Receive Location

How to Configure Tracking for a Pipeline

This topic describes how to use the BizTalk Server Administration to configure tracking for a pipeline. You might want to configure tracking for troubleshooting and auditing purposes.

You can view message properties, port events, and message events in the Find Message and Results views of Health and Activity Tracking (HAT). You can also track message events and port events for messages in the Message Flow view of HAT. For more information about HAT, see Health and Activity Tracking.

You can configure tracking for one of the default pipelines included with BizTalk Server 2006 or a custom pipeline that has been deployed into a BizTalk application. The tracking settings that you configure apply to all of the instances of the pipeline.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure tracking for a pipeline

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 and expand the BizTalk group containing the pipeline for which to configure tracking.

3. Do one of the following:

- To configure tracking for one of the default BizTalk pipelines, expand <All Artifacts>.

- To configure tracking for a custom pipeline that has been deployed into a BizTalk application, expand the application containing the pipeline.

4. Click the **Pipelines** folder, right-click the pipeline, and then click **Tracking**.

5. Configure tracking options you want, as described in the following table, and then click **OK**.

Use this	To do this
Port start and end events	Select this check box to track only when an instance starts and ends. Details include item name, assembly, and other metadata.
Message send and receive events	Select this check box to track message send and receive events. This check box is available only if **Port start and end events** is selected.
Messages before pipeline processing	Select this check box to save and track the message bodies received by the pipeline, which holds metadata such as URLs and promoted properties. If this is a receive pipeline, the message body is the raw message as submitted to the pipeline by the transport component. Depending on the application, the message might be encrypted, signed, or encoded. This check box is available only if **Message send and receive events** is selected.
Messages after pipeline processing	Select this check box to save and track the message bodies sent by the pipeline, which holds metadata such as URLs and promoted properties. If this is a receive pipeline, the message body is the processed message to be submitted to the MessageBox database, which may be XML depending on your application. This check box is available only if **Message send and receive events** is selected.

How to Configure Per-instance Pipeline Properties for a Send Port

This topic describes how to use the BizTalk Server Administration console to configure pipeline properties for a send port after the pipeline has been deployed into a BizTalk group. Changes that you make overwrite the default pipeline properties for this send port only, so if you want, you can configure different pipeline properties for each send port in the BizTalk group.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure per-instance pipeline properties for a send port

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 containing the send port for which you want to configure pipeline properties, expand Applications, and then expand the application containing the send port.

3. Click the **Send Ports** folder, right-click the send port, and then click **Properties**.

4. Click the ellipsis (...) to the right of the **Send Pipeline** box.

5. Configure the properties you want, and then click **OK**. Click Help on the properties page for more information.

6. If this is a solicit-response send port, click the ellipsis (...) to the right of the **Receive Pipeline** box.

7. Configure the properties you want, and then click **OK** twice. Click Help on the properties page for more information

How to Configure Per-instance Pipeline Properties for a Receive Location

This topic describes how to use the BizTalk Server Administration console to configure pipeline properties for a receive location after the pipeline has been deployed into a BizTalk group. Changes that you make overwrite the default pipeline properties for this receive location only, so if you want, you can configure different pipeline properties for each receive location in the BizTalk group.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To configure per-instance pipeline properties for a receive location

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 containing the receive location for which to configure pipeline properties, expand Applications, and then expand the application containing the receive location.

3. Click the **Receive Locations** folder, right-click the receive location, and then click **Properties**.

4. Click the ellipsis (...) to the right of the **Receive Pipeline** box.

5. Configure the properties you want, and then click **OK**. For more information, click **Help** on the properties page.

6. If this is a request-response receive location, click the ellipsis (...) to the right of the **Send Pipeline** box.

7. Configure the properties you want, and then click **OK** twice. For more information, click **Help** on the properties page.

Managing Resources

The topics in this section provide instructions on how to use the BizTalk Server Administration console or the BTSTask command-line tool to manage BizTalk Server resources after they have been deployed into a BizTalk group. Resources include the following types of artifacts:

- BizTalk Assemblies
- Pre- and post-processing scripts
- .NET assemblies
- COM components
- Certificates
- Ad hoc files
- BAM definitions
- Binding files
- Virtual directories

In This Section

- Managing BizTalk Assemblies
- Managing Pre- and Post-processing Scripts
- Managing .NET Assemblies, Certificates, and Other Resources
- How to Refresh a Resource

Managing BizTalk Assemblies

This section provides instructions on using the BizTalk Server Administration console or the BTSTask command-line tool to manage BizTalk assemblies after they have been deployed into a BizTalk group. A BizTalk assembly is a Microsoft Windows DLL file that contains resource information, such as orchestrations, pipelines, schemas, and maps to be used in a BizTalk Server business solution. For background information on BizTalk assemblies, see BizTalk Assemblies.

The developer uses Visual Studio 2005 to develop BizTalk assemblies, as described in Developing BizTalk Server Applications , and can deploy them from Visual Studio into BizTalk Server as described in Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application. The developer can also manage BizTalk assemblies during the development process by using either the administration console, as described in this section, or BizTalk Explorer, as described Managing Assemblies Using BizTalk Explorer .

In This Section

- How to Add a BizTalk Assembly to an Application
- How to Modify the Deployment Options of a BizTalk Assembly
- How to View the Dependencies for a BizTalk Assembly
- How to Remove a BizTalk Assembly from an Application

How to Add a BizTalk Assembly to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a BizTalk assembly to an application.

When adding the BizTalk assembly to an application, bear in mind the following important points:

- If you want to add an assembly and overwrite an assembly with the same locally unique identifier (LUID) that already exists in the application, specify the Overwrite option. If not specified, and an assembly that has the same LUID as the assembly being added already exists in the application the operation will fail. The

LUID includes the assembly file name, version, culture, and public key token. You can view the LUIDs for the artifacts in an application by using the ListApp Command.

- If the assembly you are adding has a dependency on another artifact that is not included in the application, the add operation will fail.

- When you add a BizTalk assembly, you can specify one or more of the following options for installing the assembly to the global assembly cache (GAC):

  - **Add to the global assembly cache on add resource (gacutil).** When you select this option, the assembly is installed in the GAC on the local computer when the assembly is added to an application, as a result of using the procedures in this topic.

  - **Add to the global assembly cache on MSI file import (gacutil).** When you select this option, if the application is exported to an .msi file, and the .msi file is imported into a BizTalk group, the assembly is installed in the GAC on the local computer as part of the import process.

  - **Add to the global assembly cache on MSI file install (gacutil).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, the assembly is installed in the GAC on the local computer as part of the installation process.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a BizTalk assembly to an application

Using the BizTalk Server Administration console

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 and the BizTalk Group containing the application to which you want to add the BizTalk assembly.

3. Expand Applications and the application to which you want to add a BizTalk assembly.

4. Right-click **Resources**, point to **Add** and then click **BizTalk Assemblies**.

5. Click **Add**, select the BizTalk assembly file, and then click **Open**.

6. In **Destination**, type the full path of the location where the assembly file is to be copied when the application is installed from the .msi file, including the file name. If not provided, the assembly file is not copied to the local file system during installation.

7. In **Options**, specify the options for installing the BizTalk assembly to the GAC, and then click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [**/ApplicationName:***value*] **/Type:System.BizTalk:BizTalkAssembly** [**/Overwrite**] **/Source:***value* [**/Destination:***value*] [**/Options:GacOnAdd**|**GacOnInstall**|**GacOnImport**] [**/Server:***value*] [**/Database:***value*]

How to Modify the Deployment Options of a BizTalk Assembly

This topic describes how to use the BizTalk Server Administration console to modify the deployment options of a BizTalk assembly.

You can specify the following deployment options:

- **Add to the global assembly cache on add resource (gacutil).** When you select this option, the assembly is added to the global assembly cache (GAC) on the local computer when the assembly is added to the application. In addition, if you later refresh the assembly (right-click it and click **Refresh**), the assembly is added to the GAC. Clearing the check box for this option does not remove the assembly from the GAC, if it currently exists there.

- **Add to the global assembly cache on MSI file import (gacutil).** Install the assembly to the GAC on the local computer when the application .msi file is imported.

- **Add to the global assembly cache on MSI file install (gacutil).** Install the assembly to the GAC on the local computer when the application is installed from the .msi file.

- **Destination location:** Path to which the assembly file will be copied when the application is installed. If a path is not provided, the assembly file is not copied to the local file system on installation.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. In addition, if you select an option that immediately adds the assembly to the GAC, your account must also be a member of the local Administrator's group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To modify the deployment options of a BizTalk assembly

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 containing the BizTalk assembly for which to modify deployment options, and then expand the application containing the BizTalk assembly.

3. Click the **Resources** folder, right-click the BizTalk assembly, and then click **Modify**.

4. In **Options**, select the check boxes of the deployment options that you want.

5. If necessary, in **Destination location** edit the path where the assembly will be copied when the application is installed, and then click **OK**.

to View the Dependencies for a BizTalk Assembly

This topic describes how to use the BizTalk Server Administration console to view the list of artifacts that have dependencies on a BizTalk assembly. For background information about dependencies and how they affect application deployment, see Dependencies and Application Deployment.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group or BizTalk Operators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To view the dependencies of a BizTalk assembly

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 containing the BizTalk assembly for which you want to view dependencies, and then expand the application containing the BizTalk assembly.

3. Click the **Resources** folder, right-click the BizTalk assembly, and then click **Modify**.

4. Click the **Dependencies** tab.

   The name and status of the artifacts that have dependencies on this BizTalk assembly are displayed in the list.

How to Remove a BizTalk Assembly from an Application

This topic describes how use the BizTalk Server Administration console or the command line to remove a BizTalk assembly from a BizTalk application. When you do this, the assembly and the artifacts that it includes, such as orchestrations, schemas, and pipelines, are removed from the application and the BizTalk Management database.

Before removing a BizTalk assembly, bear in mind the following important points:

- When you remove a BizTalk assembly, the assembly file is not automatically removed from the global assembly cache (GAC) or the local file system, if it exists there. You must manually remove it. For instructions, see How to Uninstall an Assembly from the GAC and How to Remove Other Files and Settings for a BizTalk Application.

- If you remove a BizTalk assembly that includes a pipeline, any send ports in the same application that use the pipeline will be reset to use the default, PassThruTransmit pipeline.

- You cannot remove a BizTalk assembly on which other artifacts depend. You must first remove the dependent artifacts. Then you can remove the BizTalk assembly.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a BizTalk assembly from an application

Using the BizTalk Server Administration console

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 containing the BizTalk assembly to remove, and then expand the application containing the BizTalk assembly.

3. Click the **Resources** folder, right-click the BizTalk assembly, and then click **Remove**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource [**/ApplicationName:***value*] **/Luid:***value* [**/Server:***value*] [**/Database:***value*]

Managing Pre- and Post-processing Scripts

This section provides instructions on using the BizTalk Server Administration console or the BTSTask command-line tool to manage pre- and post-processing scripts. You can create a script that you want to have run when a BizTalk application is imported, installed, or uninstalled (removed), and then add the script to the application. When you add a script, you specify whether it is a pre-preprocessing script, which runs before import, installation, or uninstallation starts, or a post-processing script, which runs after import, installation, or uninstallation completes.

If you select the script when exporting the application, the script is included in the .msi file for the application. When you install, uninstall, or import the application, the script automatically runs, depending on how you have written the script. For more information about creating and using scripts, see Using Pre- and Post-processing Scripts to Customize Application Deployment.

In This Section

- How to Add a Pre- or Post-processing Script to an Application
- How to Change the Destination Location of a Pre- or Post-processing Script
- How to Remove a Pre- or Post-processing Script from an Application

How to Add a Pre- or Post-processing Script to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a pre- or post-processing script to an application. When you add a script to an application, the script is included in the application .msi file, and runs when the application is imported, installed, or uninstalled.

When you add a script, you must specify whether it is a pre-preprocessing script, which will run before the application import, installation, or uninstallation starts, or a post-processing script, which will run after the application import, installation, or uninstallation completes.

You can also remove a script from an application. For instructions, see How to Remove a Pre- or Post-processing Script from an Application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a script to an application

Using the BizTalk Server Administration console

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

2. Expand the BizTalk group, expand Applications, and then right-click the folder of the application to which you want to add a script.

3. Point to **Add** and click either **Pre-processing Scripts**, if you want the script to run before application import, installation, or removal begins, or **Post-processing Scripts**, if you want the script to run after application import, installation, or uninstallation completes.

4. Click **Add** and browse to the script file to add.

5. Select the script file and click **Open**.

6. If you want to overwrite a script file that already exists in the application, select the **Overwrite all** check box. The script file must have the same file name as the one being added to be overwritten. Otherwise the new script will be added, and the existing script will remain in the application unchanged.

7. Click the **File type** drop-down list and click the file type – either **System.BizTalk:PreprocessingScript** or **System.BizTalk:PostprocessingScript**.

8. If necessary, in **Destination location** type the path where you want the script file to be copied when the application is installed, and then click **OK**. The default path will install the script to the application installation folder (%BTAD_InstallDir%).

The script is added to the application and is displayed in the Resources folder of the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [**/ApplicationName:***value*] **/Type:System.BizTalk:PreProcessingScript|System.BizTalk:PostProcessingScript** [**/Overwrite**] **/Source:***value* [**/Destination:***value*] [**/Server:***value*] [**/Database:***value*] [**/Property:Args="***argument list***"**]

How to Change the Destination Location of a Pre- or Post-processing Script

This topic describes how to use the BizTalk Server Administration console to change the destination location of a pre- or post-processing script. This is the location to which the script is copied when the application is installed. You might want to change the destination location when deploying an application into a different environment.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To modify the deployment properties of a pre- or post-processing script

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 containing the script to modify, and then expand the application containing the script.

3. Click the **Resources** folder, right-click the script, and then click **Modify**.

4. In **Destination location**, type the full path of the destination location, including the file name, and then click **OK**. (You can use the environment variable %BTAD_InstallDir% in the path to specify the application installation folder.)

How to Remove a Pre- or Post-processing Script from an Application

This topic describes how to use the BizTalk Server Administration console or the command line to remove a pre- or post-processing script from an application. This removes the script from the BizTalk Management database, so that it will not be exported in the application .msi file. This does not remove the script from the local file system, if it exists there.

If the application containing the script has been installed on the local file system, and the script is designed to run during uninstallation, you must remove the script from the file system to prevent it from running when the application is uninstalled.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a script from an application

Using the BizTalk Server Administration console

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 containing the script to remove, and then expand the application containing the script.

3. Click the **Resources** folder, right-click the script, and then click **Remove**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource [**/ApplicationName:***value*] **/Luid:***value* [**/Server:***value*] [**/Database:***value*]

Managing .NET Assemblies, Certificates, and Other Resources

This section provides instructions on using the BizTalk Server Administration console and the BTSTask command-line tool to manage the following types of resource artifacts for a BizTalk application. These resource artifacts cannot be automatically deployed into a BizTalk application from Visual Studio, so you must add them manually to the application. Once added to an application, however, you can import, export, and install them as a unit with the application and its other artifacts.

- **Files.** You can include ad hoc files, such as a Readme file. When you install the application, files are copied to the installation folder.

- **Certificates.** You can add a certificate file to an application so that you can transport the certificate from one BizTalk group to another, packaged with an application. You assign certificates to send ports and receive locations to verify identities and to establish secure links.

- **COM and COM+ components.** You can include managed COM and managed COM+ components to provide additional functionality in a BizTalk application.

- **.NET assemblies.** You can include .NET assemblies that are not specifically BizTalk assemblies in a BizTalk application. (BizTalk assemblies are .NET assemblies that contain BizTalk orchestrations, pipelines, schemas, or maps.)

- **BAM definition files.** To make BAM deployment easier, you can create a BizTalk application and add BAM definitions to it. You can then use the BizTalk application deployment features to deploy the BAM definitions into different environments.

- **Binding files.** You can add binding files to an application to make moving an application from one deployment environment to another quicker and easier.

- **Virtual directories.** You can add virtual directories to your application so that they will be deployed with the application.

In This Section

- How to Add a File to an Application

- How to Add a Certificate to an Application

- How to Add a COM Component to an Application

- How to Add a .NET Assembly to an Application

- How to Add a BAM Definition File to an Application

- How to Add a Binding File to an Application

- How to Add a Virtual Directory to an Application

- How to Modify the Deployment Options of a .NET Assembly, Certificate, or Other Resource Artifact

- How to Remove a .NET Assembly, Certificate, or Other Resource Artifact from an Application

How to Add a File to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a file to a BizTalk application. Files that you add to your application are copied to the installation folder when the application is installed. Files can also can be exported into the application's .msi file and moved to different deployment environments with the application.

If you want to overwrite a file that already exists in the application, specify the Overwrite option. The overwrite option only works when both files have the same file name. If not specified, and a file already exists in the application with the same file name as the one being added, the add operation will fail.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a file to an application

Using the BizTalk Server Administration console

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 and the BizTalk Group containing the application to which you want to add the file.

3. Expand Applications and the application to which you want to add a file.

4. Right-click the **Resources** folder, point to **Add**, and then click **Resources**.

5. Click **Add**, select the file, and then click **Open**.

6. In the **File type** drop-down list, select **System.BizTalk:File**.

7. In **Destination**, type the full path of the location where the file is to be copied when the application is installed from the .msi file, including the file name. The default is %BTAD_InstallDir%, the application installation folder. If this path is not provided, the file is not copied to the local file system during installation.

8. When finished, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [/ApplicationName:*value*] /Type:System.BizTalk:File [/Overwrite] /Source:*value* [/Destination:*value*] [/Server:*value*] [/Database:*value*]

How to Add a Certificate to an Application

This topic describes how to use the command line to add a certificate to a BizTalk application. You add a certificate to a BizTalk application so that you can transport the certificate from one BizTalk group to another, packaged with an application. You use certificates to verify identities and to establish secure links for send ports and receive locations. For more information, see How to Assign a Certificate to a Send Port and How to Assign a Certificate to a Receive Location.

When adding a certificate to an application, bear in mind the following important points:

- When you add a certificate to an application, the certificate is added to the BizTalk Management database as a certificate artifact. When you install the application, the certificate is imported into the Other People certificate store on the local computer, so you may not need to take the additional step of importing it into this store before you can assign it to a send port or receive location. When you use BTSTask to add the certificate, the certificate must exist in the Other People certificate store, and you must specify its thumbprint.

- As a best practice, if a certificate will be used by a send port or receive location in two or more applications, you should deploy the certificate in a separate application, and then reference this application from the applications that need to use the certificate. This is because only one certificate having a particular thumbprint can exist in the BizTalk group, so you will not be able to import the same certificate in two different applications. If you attempt to import two applications that each use the same certificate, the first import will succeed, and the second will not. In this case, using the Overwrite import option does not correct the problem, as the existing certificate that you want to overwrite is contained in another application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a certificate to an application

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [/ApplicationName:*value*] /Type:System.BizTalk:Certificate [/Overwrite] /Thumbprint:"*value*" [/Server:*value*] [/Database:*value*]

How to Add a COM Component to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a COM component to a BizTalk application:

When adding a COM component to an application, bear in mind the following important points:

- If you want to overwrite a COM component that already exists in the application, specify the Overwrite option. The overwrite option is required only when both artifacts have the same locally unique identifier (LUID). If not specified, and a COM component already exists in the application with the same LUID as the one being added, the add operation will fail. You can view the LUIDs for the artifacts in an application by using the ListApp Command.

- BizTalk Server does not check the dependencies for COM components to verify that they are present, so you should verify that any artifacts on which the component depends are present.

- If you add a 64-bit unmanaged COM or COM+ component, and you attempt to install the application that includes the COM or COM+ component on a 32-bit computer, the component will not be installed. It will install only on a 64-bit computer.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a COM component to an application

Using the BizTalk Server Administration console

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, expand Applications, and then expand the application to which you want to add a COM component.

3. Right-click the **Resources** folder, point to **Add**, and then click **Resources**.

4. Click **Add**, select the COM component, and then click **Open**.

5. In the **File type** drop-down list, click **System.BizTalk:Com**.

6. In **Options**, select or clear the **Register the file at destination (regsvr32)** check box according to whether or not you want the component to be added to the Windows Registry when the application is installed.

7. In **Destination**, type the full path of the location where the COM component is to be copied when the application is installed from the .msi file, including the file name. If this path is not provided, the file is not copied to the local file system during installation. This path is required if you selected the **Register the file at destination (regsvr32)** check box in the previous step.

   Example: **%BTAD_InstallDir%\MyComponent.dll**

8. When finished, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [/ApplicationName:*value*] /Type:System.BizTalk:Com [/Overwrite] /Source:*value* [/Destination:*value*] [/Options:Regsvr32OnInstall] [/Server:*value*] [/Database:*value*]

How to Add a .NET Assembly to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a .NET assembly that is not a BizTalk assembly to a BizTalk application. When adding a .NET assembly to an application, bear in mind the following important points:

- If you want to overwrite an assembly that already exists in the application, specify the Overwrite option. The overwrite option is necessary only when both assemblies have the same LUID. If not specified, and an assembly already exists in the application with the same LUID as the assembly being added, the add operation will fail. You can view the LUIDs for the artifacts in an application by using the ListApp Command.

- If the assembly you are adding has a dependency on another assembly that is not included in the application, the add operation will fail.

- When you add a .NET assembly, you can specify one or more of the following options for installing the assembly to the global assembly cache (GAC):

- **Add to the global assembly cache on add resource (gacutil).** When you select this option, the assembly is installed in the GAC on the local computer when the assembly is added to an application, as a result of using the procedures in this topic.

- **Add to the global assembly cache on MSI file import (gacutil).** When you select this option, if the application is exported to an .msi file, and the .msi file is imported into a BizTalk group, the assembly is installed in the GAC on the local computer as part of the import process.

- **Add to the global assembly cache on MSI file install (gacutil).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, the assembly is installed in the GAC on the local computer as part of the installation process.

- **Make visible to COM components (regasm).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, a managed COM assembly is added to the Windows registry on the local computer as part of the installation process. If you specify this option, you must also specify a location for the file in Destination.

- **Register serviced components (regsvcs).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, a managed COM+ assembly is added to the Windows registry on the local computer as part of the installation process. If you specify this option, you must also specify a location for the file in Destination.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a .Net assembly to an application

Using the BizTalk Server Administration console

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, expand Applications, and then expand the application to which you want to add the .NET assembly.

3. Right-click the **Resources** folder, point to **Add**, and then click **Resources**.

4. Click **Add**, click the assembly, and then click **Open**.

5. In the **File type** drop-down list, select **System.BizTalk:Assembly**.

6. In **Options**, select the deployment options for this assembly.

7. In **Destination**, type the full path of the location where the file is to be copied when the application is installed from the .msi file, including the file name. If this path is not provided, the file is not copied to the local file system during installation.

   Example: **C:\My Applications\My Files**

8. Click the **Dependencies** tab and view the artifacts on which this assembly depends.

9. If an artifact on which this assembly depends is not present in this application, and you want to add it, click **Add to Application**, browse to the artifact, and then click **Open**.

10. When finished, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table.

   BTSTask AddResource [/ApplicationName:*value*] /Type:System.BizTalk:Assembly [/Overwrite] /Source:*value* [/Destination:*value*] [/Options:GacOnAdd|GacOnInstall|GacOnImport|RegasmOnInstall|RegsvcsOnInstall] [/Server:*value*] [/Database:*value*]

How to Add a BAM Definition File to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a BAM definition (.bdf) file to a BizTalk application. Adding a BAM definition file does not deploy the BAM definition. The BAM definition is deployed when the application .msi is imported.

If you want to overwrite a BAM definition file that already exists in the application, specify the overwrite option. The overwrite option is required only when the existing BAM definition files has the same file name as the one you want to add. If not specified, and BAM definition file already exists in the application with the same file name as the one being added, the add operation will fail.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a BAM definition file to an application

Using the BizTalk Server Administration console

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, expand Applications, and then expand the application to which you want to add a BAM definition file.

3. Right-click the **Resources** folder, point to **Add**, and then click **Resources**.

4. Click **Add**, select the file of the BAM definition file, and then click **Open**.

5. In the **File type** drop-down list, select **System.BizTalk:BAM**.

6. In **Destination**, type the full path of the location where the file is to be copied when the application is installed from the .msi file, including the file name. If this path is not provided, the file is not copied to the local file system during installation, but it will be deployed when the application .msi file is imported.

   Example: **C:\My Applications\MyDefinition.bdf**

7. When finished, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [**/ApplicationName:***value*] **/Type:System.BizTalk:Bam** [**/Overwrite**] **/Source:***value* [**/Destination:***value*] [**/Server:***value*] [**/Database:***value*]

How to Add a Binding File to an Application

This topic describes how to use the BizTalk Server Administration console or the command line to add a binding file to a BizTalk application. You might want to do this to make application or assembly deployment easier, as described in Binding Files and Application Deployment.

You can export bindings into an .xml file from a BizTalk application for an assembly, application, or group, as described in Exporting Bindings, and then use one of the procedures in this topic to add the binding file to an application.

When you do this, the binding file is added to the BizTalk Management database and displays in the BizTalk Server Administration console, in the Resources folder of the application. Unlike importing a binding file, adding a binding file does not immediately apply its bindings. Instead, the bindings are applied when the application is imported into another BizTalk group.

When you add a binding file to an application, you can specify a value for the target deployment environment with a string that represents the environment, such as Test or Production. You can use any string for this value. Then, when you import the application, you can select which binding file to apply by providing the value that was specified for its target environment. When you do this, the bindings are applied from the binding file. Any existing bindings in the application that have the same name as bindings in the file are automatically overwritten.

When you import an application, bindings are applied in the following order. As bindings are applied during the import process, bindings that have already been applied are overwritten by new bindings that have the same name. In other words, the last binding of a particular name to be applied takes effect.

1. Application bindings generated by BizTalk Server that were not explicitly added to the application via a binding file, but that were explicitly selected by the user for export into the application .msi file.

2. Binding files that have been explicitly added, and do not have a target deployment environment specified. Bindings in this set are applied in no specific order.

3. Bindings that have been explicitly added, and that have an associated target deployment environment that matches the deployment environment selected for application import. Bindings in this set are applied in no specific order.

For more information about importing applications and applying bindings, see How to Import a BizTalk Application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a binding file to an application

Using the BizTalk Server Administration console

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 and the BizTalk group containing the application to which you want to add a binding file.

3. Expand Applications, and right-click the application to which you want to add a binding file.

4. Point to **Add**, and then click **Resources**.

5. Click **Add**, select the file to add, and then click **Open**.

6. To overwrite an existing binding file in this application that has the same file name, select the **Overwrite all** check box. If another file exists with the same name, and you do not select this check box, the add operation will fail.

7. In the **File type** drop-down list, select **System.BizTalk:BizTalkBinding**.

8. In **Target Environment**, type a string to represent the target deployment environment where you want the bindings in this file to be applied, such as Test, and click **OK**.

9. The binding file is added, and it displays in the Resources folder of the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [/ApplicationName:*value*]
   /Type:System.BizTalk:BizTalkBinding [/Overwrite] /Source:*value*
   [/Destination:*value*] /Property:TargetEnvironment=*value*[/Server:*value*]
   [/Database:*value*]

How to Add a Virtual Directory to an Application

This topic describes how to use the command line to add a virtual directory to a BizTalk application. You might want to do this if you have written a custom Web service or created an ASP.NET Web site for interfacing with BizTalk Server and want to deploy the virtual directory with the application.

Another way to add a virtual directory to an application is by specifying a virtual directory for a SOAP or HTTP receive location, as described in **Configuring an HTTP Receive Location**. In all cases, the virtual directory is added to the BizTalk Management database. When you add a virtual directory by using the command line, it also displays in the BizTalk Server Administration console, in the Resources folder of the application to which you added it as well as the list of artifacts in the application when you use the ListApp command. In addition, if you export the application and then import it into another BizTalk group, the virtual directory will display in the Resources folder.

When adding a virtual directory to an application, bear in mind the following points:

- When using the command line, you can overwrite a virtual directory that already exists in the application by specifying the overwrite option. The overwrite option is required only when the existing virtual directory has the same name as the one you want to add. If not specified, and a virtual directory already exists in the application with the same name as the one being added, the add operation will fail.

- If you add a virtual directory from a 64-bit version of the Web service, and you attempt to install the application that includes the virtual directory on a 32-bit computer, the virtual directory will not be installed. It must be installed on a 64-bit computer.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a virtual directory to an application

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [/ApplicationName:*value*] /Type:System.BizTalk:WebDirectory [/Overwrite] /Source:*value* [/Destination:*value*] [/Server:*value*] [/Database:*value*]

How to Modify the Deployment Options of a .NET Assembly, Certificate, or Other Resource Artifact

This topic describes how to use the BizTalk Server Administration console to modify the deployment options of the following resource artifacts:

- .NET assemblies
- COM components
- Certificates
- Ad hoc files
- BAM definitions
- Virtual directories

You might want to modify the deployment properties to change the destination location to which the artifact file will be copied when the application is installed on the local computer. If you do not provide a path, the file will not be copied to the local computer on installation.

In addition, you might want to modify the following options for a .NET assembly:

- **Add to the global assembly cache on add resource (gacutil).** When you select this option, the assembly is added to the global assembly cache (GAC) on the local computer when the assembly is added to the application. In addition, if you later refresh the assembly (right-click it and click **Refresh**), the assembly is added to the GAC. Clearing the check box for this option does not remove the assembly from the GAC, if it currently exists there.

- **Add to the global assembly cache on MSI file import (gacutil).** When you select this option, if the application is exported to an .msi file, and the .msi file is imported into a BizTalk group, the assembly is installed in the GAC on the local computer as part of the import process.

- **Add to the global assembly cache on MSI file install (gacutil).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, the assembly is installed in the GAC on the local computer as part of the installation process.

- **Make visible to COM components (regasm).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, a managed COM assembly is added to the Windows registry on the local computer as part of the installation process. If you specify this option, you must also specify a location for the file in Destination.

- **Register serviced components (regsvcs).** When you select this option, if the application is exported to an .msi file, and the application is installed on a computer from the .msi file, a managed COM+ assembly is added to the Windows registry on the local computer as part of the installation process. If you specify this option, you must also specify a location for the file in Destination.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. In addition, if you select an option that immediately adds the assembly to the GAC, your account must also be a member of the local Administrator's group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To modify the deployment properties of a resource artifact

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 containing the artifact for which to modify deployment options, and then expand the application containing the artifact.

3. Click the **Resources** folder, right-click the artifact, and then click **Modify**.

4. On the **Options** tab, modify the deployment options as necessary, and then click **OK**.

How to Remove a .NET Assembly, Certificate, or Other Resource Artifact from an Application

This topic describes how to use the BizTalk Server Administration console or the command line to remove the following resource artifacts from a BizTalk application. Using the procedures in this topic removes the artifact from the BizTalk Management database. It does not remove the artifact from the file system, certificate store, Internet Information Services, or the Windows registry, if it exists in any of these locations. In addition, if you remove a binding file, the bindings remain unchanged – only the binding file is removed.

- .NET assemblies
- COM components
- Certificates
- Ad hoc files
- BAM definitions

- Binding files
- Virtual directories

When you edit an HTTP receive location to point to a different virtual directory, the virtual directory is added to the list in the Resources folder, but is not stored in the BizTalk Management database. If a virtual directory is not stored in the BizTalk Management database, you cannot remove it by using the procedures in this topic. When you export the application .msi file containing the virtual directory and then import it into a different group (or into the same group after deleting the application), the virtual directory will be added to the BizTalk Management database for that group.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a resource artifact from an application

Using the BizTalk Server Administration console

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 containing the resource artifact to remove, and then expand the application containing the artifact.
3. Click the **Resources** folder, right-click the artifact, and then click **Remove**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.
2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource /ApplicationName:*value* **/Luid:***value*
   [**/Server:***value*] [**/Database:***value*]

How to Refresh a Resource

This topic describes how to use the BizTalk Server Administration console to refresh a resource artifact. This updates the artifact information in the BizTalk Management database.

Resource artifacts are contained in an application's Resources folder. You might want to refresh a resource artifact if you make changes to the source file and want to replace the file in the application with the updated file.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To refresh a resource artifact

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 containing the resource artifact to refresh, and then expand the application containing the artifact.

3. Click the **Resources** folder, right-click the file to refresh, and then click **Modify**.

4. In the Modify Resources dialog box, select the file to refresh and click **Refresh**.

5. Browse to the updated file with which you want to replace the current file, and then click **OK**.

   The current file is replaced with the updated file. In addition, the configuration, settings displayed on the Options tab of the Modify Resources dialog box are applied to the refreshed file. For more information about changing these settings, click **Help** in the dialog box.

Managing BAS Artifacts

This section provides instructions on using the BizTalk Server Administration console to add Business Activity Services (BAS) artifacts to BizTalk applications, and remove them from BizTalk applications. The BAS artifacts you can add to BizTalk applications include the following:

- **Self profiles**. A self profile contains information about your company that you want to share with partners. For more information about self profiles, see Creating Profiles for Trading Partners [BTS2006].

- **Partner profiles**. A partner profile contains information about a trading partner. Partner profiles include non-technical information such as addresses and contacts, and technical information such as send ports and certificates your organization uses to exchange messages with the partner. For more information about partner profiles, see Creating Profiles for Trading Partners [BTS2006].

- **Partner groups**. A partner group contains partner profiles. You use a partner group to manage relationships with multiple trading partners. You add partner profiles that you want to manage together to a group. For more information about partner groups, see Organizing Partner Profiles with Partner Groups .

- **Agreements**. Agreements represent the relationship between partners based on business processes. You can create an agreement between a self profile for your organization and a trading partner profile or partner group. You can associate an agreement with a partner group to apply it to all partner profiles in the group. For more information about agreements, see Establishing Partner Relationships with Agreements .

- **Templates**. InfoPath solution templates define the documents exchanged by partners in a business transaction, such as a purchase order or an invoice. A solutions developer creates InfoPath solution templates, and imports the XML schemas used in the orchestrations to the BAS site. For more information about templates, see Business Document Schemas and Templates Creation .

Additionally, you can use the BizTalk Server Administration console to configure BAS template properties. For information about creating BAS artifacts, see Managing Partner Relationships with BAS.

When you create BAS artifacts in the BAS site, the artifacts are added to the Trading Partner Management (TPM) database. You can then use the administration console to add the BAS artifacts to BizTalk applications.

In This Section

- How to Add a Self Profile
- How to Add a Partner Profile
- How to Add a Partner Group
- How to Add an Agreement
- How to Add a Template
- How to Change Template Properties
- How to Remove a Self Profile
- How to Remove a Partner Profile
- How to Remove a Partner Group
- How to Remove an Agreement
- How to Remove a Template

How to Add a Self Profile

This topic describes how to use the BizTalk Server Administration console or the command line to add a BAS self profile to a BizTalk application. You can only add self profiles that exist on the BAS Web.

When you use the BAS site to create BAS artifacts, such as self profiles, the artifacts are added to the Trading Partner Management (TPM) database. You can then use the administration console to add the BAS artifacts to BizTalk applications. Depending on your BizTalk Server configuration, the TPM database may exist as a database, or it may exist as a set of tables in the BizTalk Management database.

Occasionally, the BAS site, the TPM database, and the BizTalk Administration console get out of sync. Therefore, we recommend that you refresh all three periodically. For information about synchronizing the BAS site with the TPM and BizTalk Management databases, see Managing Partner Data . To refresh the BizTalk Administration console, right click the **My Profiles** node, and click **Refresh**.

If you want to overwrite a self profile that already exists in the application, specify the overwrite option. The overwrite option is required only when the existing self profile has the same name as the one you want to add. If the overwrite option is not specified, and a self profile already exists in the application with the same name as the one being added, the add operation fails.

Keep in mind the following considerations when you add a self profile to an application:

- Before you can add a self profile to an application, the self profile must exist in the BizTalk Management database for the BizTalk group. For information about creating a self profile, see How to Create a Profile .

- The self profile cannot already exist in another application in the BizTalk group.

- Self profiles do not have any dependency artifacts. Self profiles may be dependency artifacts for agreements. If you add an agreement associated with a self profile, you have the option to add the associated self profile at the same time.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a self profile to an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application to which you want to add a self profile, expand **BAS Artifacts**, right-click **My Profiles**, and then click **Add**.

4. In the **Add My Profiles** dialog box, do the following:

Use this	To do this
Type name or select from the list	From the list, click the self profile you want to add to the application.
Add	Click to add the selected self profile to the list of self profiles to add to the application.

5. Click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

BTSTask AddResource [/ApplicationName:"*value*"] /Name:"*value*" /Subtype:MyProfile /Type:System.BizTalk:Bas [/Overwrite] [/Server:*value*] [/Database:*value*]

How to Add a Partner Profile

This topic describes how to use the BizTalk Server Administration console or the command line to add a BAS partner profile to a BizTalk application. You can only add partner profiles that exist on the BAS Web site.

When you use the BAS site to create BAS artifacts, such as partner profiles, the artifacts are added to the Trading Partner Management (TPM) database. You can then use the administration console to add the BAS artifacts to BizTalk applications. Depending on your BizTalk Server configuration, the TPM database may exist as a database, or it may exist as a set of tables in the BizTalk Management database.

Occasionally, the BAS site, the TPM database, and the BizTalk Administration console get out of sync. Therefore, we recommend that you refresh all three periodically. For information about synchronizing the BAS site with the TPM and BizTalk Management databases, see Managing Partner Data . To refresh the BizTalk Administration console, right click the **Partner Profiles** node, and click **Refresh**.

If you want to overwrite a partner profile that already exists in the application, specify the overwrite option. The overwrite option is required only when the existing partner profile has the same name as the one you want to add. If the overwrite option is not specified, and a partner profile already exists in the application with the same name as the one being added, the add operation fails.

Keep in mind the following considerations when you add a partner profile to an application:

- Before you can add a partner profile to an application, the partner profile must exist in the BizTalk Management database for the BizTalk group. For information about creating a partner profile, see How to Create a Profile .

- The partner profile cannot already exist in another application in the BizTalk group.

- When you add a partner profile to an application, you will be prompted to add any dependent (associated) artifacts to the same application. Partner profiles are associated with send ports.

- If any of the send ports associated with the partner profile already belong to different applications, you can select to add a reference to the applications containing the send ports when you add the partner profile to an application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a partner profile to an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application to which you want to add a partner profile, expand **BAS Artifacts**, right-click **Partner Profiles**, and then click **Add**.

4. In the **Add Partner Profiles** dialog box, do the following:

Use this	To do this
Type name or select from the list	From the list, click the partner profile you want to add to the application.
Add	Click to add the selected partner profile to the list of partner profiles to add to the application.

5. Click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask AddResource [**/ApplicationName:"***value***"**] **/Name:"***value***" /Subtype:PartnerProfile /Type:System.BizTalk:Bas** [**/Overwrite**] [**/Server:***value*] [**/Database:***value*]

How to Add a Partner Group

This topic describes how to use the BizTalk Server Administration console or the command line to add a BAS partner group to a BizTalk application. You can only add partner groups that exist on the BAS Web site.

When you use the BAS site to create BAS artifacts, such as partner groups, the artifacts are added to the Trading Partner Management (TPM) database. You can then use the administration console to add the BAS artifacts to BizTalk applications. Depending on your BizTalk Server configuration, the TPM database may exist as a database, or it may exist as a set of tables in the BizTalk Management database.

Occasionally, the BAS site, the TPM database, and the BizTalk Administration console get out of sync. Therefore, we recommend that you refresh all three periodically. For information about synchronizing the BAS site with the TPM and BizTalk Management databases, see Managing Partner Data . To refresh the BizTalk Administration console, right click the **Partner Groups** node, and click **Refresh**.

If you want to overwrite a partner group that already exists in the application, specify the overwrite option. The overwrite option is required only when the existing partner group has the same name as the one you want to add, or if the partner group contains partner profiles that you have already added to the application.

If the overwrite option is not specified, and a partner group already exists in the application with the same name as the one being added, the add operation fails. Additionally, if the partner group contains partner profiles that you have already added to the application, the add operation fails.

Keep in mind the following considerations when you add a partner group to an application:

- Before you can add a partner group to an application, the partner group must exist in the BizTalk Management database for the BizTalk group. For information about creating a partner group, see How to Create a Partner Group .

- The partner group cannot already exist in another application in the BizTalk group.

- When you add a partner group to an application, you have the option to add any dependent (associated) artifacts to the same application. Partner groups are associated with partner profiles and partner groups. Note that each associated artifact has additional associated artifacts that will also be added to the application. For example, partner groups are associated with partner profiles and partner groups, and partner profiles are associated with send ports.

- If any of the partner profiles or partner groups associated with the partner group already belong to different applications, you can select to add a reference to the applications containing the artifacts when you add the partner group to an application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a partner group to an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application to which you want to add a partner group, expand **BAS Artifacts**, right-click **Partner Groups**, and then click **Add**.

4. In the **Add Partner groups** dialog box, do the following:

Use this	To do this
Type name or select from the list	From the list, click the partner group you want to add to the application.
Add	Click to add the selected partner group to the list of partner groups to add to the application.

5. Click **OK**.

6. In the message box, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

```
BTSTask AddResource [/ApplicationName:"value"] /Name:"value" /Subtype:PartnerGroup /Type:System.BizTalk:Bas [/Overwrite] [/Server:value] [/Database:value]
```

How to Add an Agreement

This topic describes how to use the BizTalk Server Administration console or the command line to add a BAS agreement to a BizTalk application. You can only add agreements that exist on the BAS Web site.

When you use the BAS site to create BAS artifacts, such as agreements, the artifacts are added to the Trading Partner Management (TPM) database. You can then use the administration console to add the BAS artifacts to BizTalk applications. Depending on your BizTalk Server configuration, the TPM database may exist as a database, or it may exist as a set of tables in the BizTalk Management database.

Occasionally, the BAS site, the TPM database, and the BizTalk Administration console get out of sync. Therefore, we recommend that you refresh all three periodically. For information about synchronizing the BAS site with the TPM and BizTalk Management databases, see Managing Partner Data . To refresh the BizTalk Administration console, right click the **Agreements** node, and click **Refresh**.

If you want to overwrite an agreement that already exists in the application, specify the overwrite option. The overwrite option is required only when the existing agreement has the same name as the one you want to add, or if the agreement contains partner profiles or partner groups that you have already added to the application.

If the overwrite option is not specified, and an agreement already exists in the application with the same name as the one being added, the add operation fails. Additionally, if the agreement is associated with partner profiles or partner groups that you have already added to the application, the add operation fails.

Keep in mind the following considerations when you add an agreement to an application:

- Before you can add an agreement to an application, the agreement must exist in the BizTalk Management database for the BizTalk group. For information about creating an agreement, see How to Create an Agreement .

- The agreement cannot already exist in another application in the BizTalk group.

- When you add an agreement to an application, you have the option to add any dependent (associated) artifacts to the same application. Agreements are associated with one self profile and one or more partner profiles or profile groups. Note that associated partner profiles and profile groups have associated artifacts that you can optionally add to the application at the same time you add the Agreement. Profile groups are associated with partner profiles, and partner profiles are associated with send ports.

- If any of the artifacts associated with the agreement already belong to different applications, you can select to add a reference to the applications containing the artifacts when you add the agreement to an application.

- In addition to adding references to any applications that contain BAS artifacts associated with the artifacts you add to this application, you must add a reference to the default application because it contains dependencies that are required to deploy the application. By default, the default application is BizTalk Application 1. For instructions on adding a reference to an application, see How to Add a Reference to Another Application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add an agreement to an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application to which you want to add an agreement, expand **BAS Artifacts**, right-click **Agreements**, and then click **Add**.

4. In the **Add Agreements** dialog box, do the following:

Use this	To do this
Type name or select from the list	From the list, click the agreement you want to add to the application.
Add	Click to add the selected agreement to the list of agreements to add to the application.

5. Click **OK**.

6. In the message box, click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

```
BTSTask AddResource [/ApplicationName:"value"] /Name:"value" /Subtype:Agreement /Type:System.BizTalk:Bas [/Overwrite] [/Server:value] [/Database:value]
```

How to Add a Template

This topic describes how to use the BizTalk Server Administration console or the command line to add a BAS template to a BizTalk application. You can only add templates that exist on the BAS Web.

When you use the BAS site to add templates, the templates are added to the Trading Partner Management (TPM) database. You can then use the administration console to add the templates to BizTalk applications. Depending on your BizTalk Server configuration, the TPM database may exist as a database, or it may exist as a set of tables in the BizTalk Management database.

Occasionally, the BAS site, the TPM database, and the BizTalk Administration console get out of sync. Therefore, we recommend that you refresh all three periodically. For information about synchronizing the BAS site with the TPM and BizTalk Management databases, see Managing Partner Data . To refresh the BizTalk Administration console, right click the **Templates** node, and click **Refresh**.

If you want to overwrite a template that already exists in the application, specify the overwrite option. The overwrite option is required only when the existing template has the same name as the one you want to add.

If the overwrite option is not specified, and a template already exists in the application with the same name as the one being added, the add operation fails.

Keep in mind the following considerations when you add a template to an application:

- Before you can add a template to an application, the template must exist in one of the BAS site document libraries. For information about creating templates, see Business Document Schemas and Templates Creation .
- The template cannot already exist in another application in the BizTalk group.
- After you add a template to an application, you can designate the document library where you want the template to be installed when the application is imported. For information, see How to Change Template Properties.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To add a template to an application

Using the BizTalk Server Administration console

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration** and the **BizTalk group**.

3. Expand **Applications**, expand the application to which you want to add a template, expand **BAS Artifacts**, right-click **Templates**, and then click **Add**.

4. In the **Add Templates** dialog box, do the following:

Use this	To do this
Type name or select from the list	From the list, click the template you want to add to the application.
Add	Click to add the selected template to the list of templates to add to the application.

5. Click **OK**.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

```
BTSTask AddResource [/ApplicationName:"value"] /Name:"value" /Subtype:Template /SourceDocLibUrl:"value" /Type:System.BizTalk:Bas [/Overwrite] [/Server:value] [/Database:value] -Property:DestinationDocLibName="value"
```

How to Change Template Properties

Use the **<Template Name> Properties** dialog box to specify the location where you want the template installed when you deploy your BizTalk application. You can only modify the properties of a template that has already been added to an application.

Keep in mind the following consideration when you add a template to an application:

- Before you can change the properties of a template, you must add the template to an application. For information about adding a template to an application, see How to Add a Template.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To change the properties of a template

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

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

3. Expand **Applications**, expand the application to which you want to add a template, expand **BAS Artifacts**, and then click **Templates**.

4. In the **Templates** node details pane, right-click the template, and then click **Properties**.

5. In the **<Template Name> Properties** dialog box, do the following:

Use this	To do this
Source document library	View the source location for the template.
Destination document library	Type the name of the document library where you want the template installed.

6. Click **OK**.

How to Remove a Self Profile

This topic describes how to use the BizTalk Server Administration console or the command line to remove a BAS self profile from a BizTalk application.

Keep in mind the following considerations when you remove a self profile from an application:

- If the self profile is shared with any other applications, removing the self profile from its application makes it unavailable to all of the referencing applications.

- If the self profile is associated with an agreement, you will not be able to remove the self profile. If you remove the agreement, you can remove the self profile at the same time.

- When you remove a self profile from an application, you have the option to delete it from the BizTalkMgmtDb database.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a self profile from an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application from which you want to remove a self profile, expand **BAS Artifacts**, and then click **My Profiles**.

4. In the **My Profiles** node details pane, right-click the self profile you want to remove, and then click **Remove**.

5. In the **Remove Resources** dialog box, click **OK** to remove the self profile from the application.

6. In the **BAS Artifacts** dialog box, do one of the following:

Use this	To do this
Yes	Click to remove the self profile and the associated party from the BizTalkMgmtDb database, and from the BAS Web site.
No	Click to only remove the self profile from the application.

7.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

```
BTSTask RemoveResource [/ApplicationName:value] /Luid:value [/Server:value] [/Database:value]
```

How to Remove a Partner Profile

This topic describes how to use the BizTalk Server Administration console or the command line to remove a BAS partner profile from a BizTalk application.

Keep in mind the following considerations when you remove a partner profile from an application:

- If the partner profile is shared with any other applications, removing the partner profile from its application makes it unavailable to all of the referencing applications.
- If the partner profile is associated with an agreement, you will not be able to remove the partner profile. If you remove the agreement, you can remove the partner profile at the same time.
- When you remove a partner profile from an application, you have the option to delete it from the BizTalkMgmtDb database.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a partner profile from an application

Using the BizTalk Server Administration console

1. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2006 Administration** and the **BizTalk group**.
3. Expand **Applications**, expand the application from which you want to remove a partner profile, expand **BAS Artifacts**, and then click **Partner Profiles**.
4. In the **Partner Profiles** node details pane, right-click the partner profile you want to remove, and then click **Remove**.
5. In the **Remove Resources** dialog box, click **OK** to remove the partner profile from the application.
6. In the **BAS Artifacts** dialog box, do one of the following:

Use this	To do this
Yes	Click to remove the partner profile and the associated party from the BizTalkMgmtDb database, and from the BAS Web site.
No	Click to only remove the partner profile from the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.
2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource [/ApplicationName:value] /Luid:value [/Server:value] [/Database:value]

How to Remove a Partner Group

This topic describes how to use the BizTalk Server Administration console or the command line to remove a BAS partner group from a BizTalk application.

Keep in mind the following considerations when you remove a partner group from an application:

- If the partner group is shared with any other applications, removing the partner group from its application makes it unavailable to all of the referencing applications.
- If the partner group is associated with an agreement, you will not be able to remove the partner group. If you remove the agreement, you can remove the partner group at the same time.
- When you remove a partner group from an application, you have the option to delete it and any members and associated artifacts from the BizTalkMgmtDb database. Partner Groups members include partner profiles and partner groups. Partner profiles are associated with send ports.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a partner group from an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application from which you want to remove a partner group, expand **BAS Artifacts**, and then click **Partner groups**.

4. In the **Partner groups** node details pane, right-click the partner group you want to remove, and then click **Remove**.

5. In the **Remove Resources** dialog box, click **OK** to remove the partner group from the application.

6. In the **BAS Artifacts** dialog box, do one of the following:

Use this	To do this
Yes	Click to remove the partner group and the associated party from the BizTalkMgmtDb database, and from the BAS Web site.
No	Click to only remove the partner group from the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

BTSTask RemoveResource [/ApplicationName:value] /Luid:value [/Server:value] [/Database:value]

How to Remove an Agreement

This topic describes how to use the BizTalk Server Administration console or the command line to remove a BAS agreement from a BizTalk application.

Keep in mind the following considerations when you remove an agreement from an application:

- If the agreement is shared with any other applications, removing the agreement from its application makes it unavailable to all of the referencing applications.

- When you remove an agreement, you can remove the associated self profile, partner profiles, and profile groups at the same time.

- When you remove an agreement from an application, you have the option to delete it from the BizTalkMgmtDb database.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove an agreement from an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application from which you want to remove an agreement, expand **BAS Artifacts**, and then click **Agreements**.

4. In the **Agreements** node details pane, right-click the agreement you want to remove, and then click **Remove**.

5. In the **Remove Resources** dialog box, click **OK** to remove the agreement from the application.

6. In the **BAS Artifacts** dialog box, do one of the following:

Use this	To do this
Yes	Click to remove the agreement and the associated party from the BizTalkMgmtDb database, and from the BAS Web site.
No	Click to only remove the agreement from the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource [/ApplicationName:value] /Luid:value [/Server:value] [/Database:value]

How to Remove a Template

This topic describes how to use the BizTalk Server Administration console or the command line to remove a BAS template from a BizTalk application.

Keep in mind the following considerations when you remove a template from an application:

- If the template is shared with any other applications, removing the template from its application makes it unavailable to all of the referencing applications.

- When you remove a template from an application, you have the option to delete it from the BizTalkMgmtDb database.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To remove a template from an application

Using the BizTalk Server Administration console

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

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

3. Expand **Applications**, expand the application from which you want to remove a template, expand **BAS Artifacts**, and then click **Templates**.

4. In the **Templates** node details pane, right-click the template you want to remove, and then click **Remove**.

5. In the **Remove Resources** dialog box, click **OK** to remove the template from the application.

6. In the **BAS Artifacts** dialog box, do one of the following:

Use this	To do this
Yes	Click to remove the template and the associated party from the BizTalkMgmtDb database, and from the BAS Web site.
No	Click to only remove the template from the application.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveResource [/ApplicationName:value] /Luid:value [/Server:value] [/Database:value]

Undeploying BizTalk Applications

You must undeploy a BizTalk application to remove it from your BizTalk group and host computers. This section describes the following tasks involved in undeploying an application:

- **Delete the BizTalk application.** This removes its data from the BizTalk databases for the BizTalk group. When you do this, the application no longer displays in the BizTalk Administration console.

- **Uninstall the BizTalk application.** This removes the files and settings associated with the application and its artifacts from the local file system.

- **Removing other files and settings.** You may also need to remove certain other files and settings associated with the application.

In This Section

- How to Delete a BizTalk Application from the BizTalk Group

- How to Uninstall a BizTalk Application

- How to Remove Other Files and Settings for a BizTalk Application

How to Delete a BizTalk Application from the BizTalk Group

You can delete an application from the BizTalk group. This removes all of its data from the BizTalk databases for the group, and the application no longer displays in the BizTalk Server Administration console. It does not uninstall the application.

Before you delete an application, bear in mind the following points:

- The application must be stopped before you can delete it. You should use the Full Stop option for stopping the application, as described in How to Start and Stop a BizTalk Application.

- You can delete an application only if no other applications contain references to it. If other applications contain references to the application you want to remove, you must first remove the references from the other applications. For instructions, see How to Remove a Reference to Another Application.

- You cannot delete an application if it contains a send port group of which a send port in another application is a member. You must unenlist the member send port before you can delete the application. For instructions, see How to Unenlist a Send Port or Send Port Group.

- You cannot delete an application if it contains a send port that is referenced by a party. You can delete the reference from the party, delete the send port, or move the send port to a different application. For instructions on performing these tasks, see How to View or Edit a Party, How to Delete a Send Port, or How to How to Move an Artifact to a Different Application.

- You cannot delete the default application. If you want to delete it, you must first make another application the default. For instructions, see How to Change the Default Application.

- You cannot delete an application if an orchestration in the application has a suspended instance.

- If the application contains BAS artifacts, when you use the administration console to delete a BizTalk application, you are given the option to remove the BAS artifacts from the BAS site as well.

- To completely undeploy a BizTalk application, you must also take the steps described in How to Uninstall a BizTalk Application, and you may also need to remove other files and settings, as described in How to Remove Other Files and Settings for a BizTalk Application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information on permissions, see Permissions Required for Deploying and Managing a BizTalk Application.

To delete a BizTalk application

Using the BizTalk Server Administration console

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 expand Applications.

3. Right-click the application folder, and then click **Delete**.

4. Click **Yes** to confirm the deletion.

   BizTalk Server deletes all of the application data from the BizTalk databases and removes the application from the administration console.

   If BizTalk Server cannot delete any of the application artifacts, the delete operation fails. In this case, BizTalk Server attempts to roll back the delete operation.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate values, as described in the following table:

   BTSTask RemoveApp /ApplicationName:*value* [**/Server:***value*] [**/Database:***value*]

How to Uninstall a BizTalk Application

This topic describes how to use the Add or Remove Programs control panel or the BTSTask command-line tool to uninstall a BizTalk application. These are the only supported methods for uninstalling an application. If you installed multiple .msi files for this application, for example to update the application, double-clicking the .msi file or using msiexec may not completely uninstall the application and are therefore not supported uninstallation methods.

A pre- or post-processing script should be included in the application .msi file to remove all of the files and settings associated with the application when it is uninstalled. If a pre- or post-processing script has not been included, the procedures in this topic will remove the application's files and settings from the local file system, with the following exceptions.

- If the application includes a virtual directory, the virtual directory and its files are deleted, unless files were added to the virtual directory after the application was installed. In this case, the virtual directory and the added files are not deleted. If you want to delete the virtual directory and the added files, you must do so explicitly.

- Pre- and post-processing scripts are deleted, but any files added by the scripts during installation or uninstallation are not deleted, nor are any actions taken by the scripts undone. If you want to delete files added by scripts or undo their actions, you must do so explicitly.

- Certificates are never deleted when you uninstall a BizTalk application. If you want to delete a certificate, you must do so explicitly. In addition, components are not removed from the Windows Registry and BizTalk assemblies are not removed from the global assembly cache (GAC). If you want to remove them, you must do so explicitly. For more information, see How to Remove Other Files and Settings for a BizTalk Application.

If you cancel the uninstallation operation before it completes, BizTalk Server will roll back the uninstallation, except for any custom actions that were taken by pre- or post-processing scripts before the operation was cancelled. You can check to see if custom actions were taken by looking in the Application view of Windows Event Viewer. If a success is logged for BizTalk Server Deployment and a failure for MSI Installer, it means that custom actions were not rolled back. To restore the application to its state before you started the uninstallation, double-click the .msi file and reinstall the application. If multiple .msi files have been installed for this application, you should double-click each .msi file to reinstall the application in the order in which the .msi files were originally installed.

For more information about post-processing scripts, see Using Pre- and Post-processing Scripts to Customize Application Deployment.

Prerequisites

To perform the procedures in this topic, you must be logged on with appropriate permissions. For more information, see Permissions Required for Deploying and Managing a BizTalk Application.

To uninstall a BizTalk application

Using Add or Remove Programs

1. On the computer running the application, click **Start**, click **Control Panel**, and then double-click **Add or Remove Programs**.

2. Click **Change or Remove Programs**, click the BizTalk application to remove, and then click **Remove**.

   Windows Installer removes the specified application.

3. If the application is running on multiple computers, repeat these steps on each computer.

Using the command line

1. Open a command prompt as follows: Click **Start**, click **Run**, type **cmd**, and then click **OK**.

2. Type the following command, substituting the appropriate value, as described in the following table:

   BTSTask UninstallApp [**/ApplicationName:***value*]