

## APPENDIX C

# Using BizTalk Native Adapters

Adapters are the components that enable the BizTalk messaging system to interface with an external communications protocol, such as HTTP, and line of business applications. As you know from Chapters 2 and 5, they mediate between the protocol and the messaging pipeline in use. Because they are protocol specific, they know how to extract the message payload from the wire format of the protocol. For example, a message sent via HTTP is text prefaced with additional text, the HTTP headers. Because they are BizTalk aware, adapters know how to add the message context metadata that other components inside BizTalk can interrogate. In Chapter 7, you learned how to write your own custom adapters.

Most integration projects employ well-known messaging protocols. Perhaps the single-most widely seen legacy pattern (and certainly the most common BizTalk sample pattern) is to use files as messages. It would be burdensome to require customers to write their own adapters for such common situations, so BizTalk ships with a collection of standard adapters. These are known as *native adapters*. If you've worked the samples throughout this book, you've encountered several of these. In this appendix, we'll look at the native adapters in more detail. The main focus will be on configuring native adapters, but we'll also pass along some observations and tips to help you in your projects. Some of the adapters have documented property schemas you can use to programmatically configure the adapters, so we'll also provide a very brief look at the BizTalk APIs for configuring native adapters from code.

### IN THIS APPENDIX

- Adapters Included in BizTalk Server 2004
- File System Adapter
- FTP Adapter
- HTTP Adapter
- SOAP Adapter
- SMTP Adapter
- MSMQT Adapter
- Base EDI Adapter
- SQL Server Adapter

**NOTE**

The source code used in this appendix is available for download at the Sams Publishing website.

---

## Adapters Included in BizTalk Server 2004

Microsoft BizTalk Server 2004 includes the following native protocol adapters:

- File
- FTP
- HTTP
- SOAP (Web services)
- SMTP (send only)
- MSMQT
- Base EDI
- SQL Server

**NOTE**

In addition, supported MSMQ, MQSeries, and WSE 2.0 (WS-Security and WS-Policy) adapters are available on the Web for download as part of your BizTalk Server 2004 license. For adapters such as ODBC, Windows Sharepoint Services have been developed by the community and are available on [gotdotnet.com](http://gotdotnet.com).

---

The file adapter addresses the most commonly encountered legacy integration situation and is also the simplest adapter to insert into an integration scenario. This is why most labs and tutorials use the file adapter so extensively. Simply stated, everyone has a file system and understands the idea of picking up a file in a known location or writing a message to a file folder at a specified location.

The FTP adapter carries this a step further to the Internet. Many systems, including mainframes and UNIX machines, that are capable of writing to files have been integrated using FTP. Adding SMTP and HTTP adapters extends coverage to the most common Internet protocols, and SOAP completes this standardization by focusing on the particular usage of HTTP by Web services.

MSMQT is new with this version of BizTalk and requires more explanation. It is not simply an adapter interfacing with Windows MSMQ (although that is also available), but rather a queuing implementation integrated with the BizTalk MessageBox that is compatible with the MSMQ wire format. We'll explain this further and examine the ramifications of this new implementation later in this chapter.

EDI is a B2B standard of sufficient standing to make it worthwhile to have an EDI-compliant adapter in the product. The native adapter for this covers the basic EDI specification. It does

not replace EDI-specific products; indeed, most implementations with a significant EDI element are better suited to implementation on top of third-party addons such as the Covast EDI accelerator.

Much business data resides in relational databases, and BizTalk is critically dependent on Microsoft SQL Server for its implementation. It is useful, therefore, to provide a native adapter for this RDBMS so that customers who have invested in SQL Server can leverage that investment by using it as a source (or sink) of messages. The SQL Server native adapter enables you to use a recordset driven through SQLXML as a message or message interchange.

The installation program for BizTalk Server 2004 installs the native adapters with default settings. The MSMQT adapter requires postinstallation configuration. The MSMQT adapter is not turned on by default as it exposes an incoming port, and leaving it off reduces the potential attack surface if you are not using this functionality. More critically, the SQL Server adapter has some preinstallation prerequisites that must be met before you install BizTalk. If you are interested in using the SQL Server adapter and you have not yet installed BizTalk, you might want to jump forward to the “SQL Server Adapter” section right now. Otherwise, we’ll work our way through the native adapters in the order we’ve previously listed them.

## File System Adapter

The file system adapter has a relatively simple task to perform and is probably the easiest adapter to use. It uses disk files as messages. A file receive adapter waits on Windows file system events on a specified folder on the local file system or a network share for the arrival of a file whose name conforms to a mask you specify. On occasions it also swaps to polling mode when Windows file system events fail. When such a file is detected, the adapter submits it to the messaging engine. If the file is successfully processed by the pipeline associated with the receive location, it is written to the MessageBox for further processing. If the file is read but fails in the pipeline, the message will be written to the suspended queue in the MessageBox. After the message is in the MessageBox, whether processed or suspended, the adapter deletes the file from disk. If the adapter is unable to write the message to the MessageBox, the disk file is not deleted.

The file adapter provides an additional and somewhat obscure safeguard for file receive locations. Given that the Windows file system is not transactional, there is a chance that duplicate messages could be generated if a receive location is shut down after the adapter has read the file but before it deletes it from disk. To guard against this, you can turn on file renaming. When file renaming is turned on, the adapter first obtains an exclusive lock on the target file, and then appends the filename with the extension `.BTS-WIP` when it commences work, but before reading the file’s contents. If the message is successfully written to the MessageBox, whether processed or suspended, the renamed file is deleted, just as the original file is deleted when renaming is turned off. If the adapter is unable to write to the MessageBox, it removes the appended extension so that the file may be processed at a later time. Renaming works as a safeguard because the file adapter will ignore any file it encounters with the `.BTS-WIP` extension, except those it has renamed itself. Renaming is one facet of the file adapter’s operation that may be controlled by a Registry key. We’ll discuss controlling renaming as well as other similar facets in the following section.

## Configuring the File Receive Adapter

If you've worked through the book to this point, you've encountered the problem of file receive adapter configuration. Indeed, your very first integration project, in Chapter 2, introduced you to this problem. Because we configured that receive location in the Orchestration Designer at design time, a better example is found in Chapter 5 on page 142. The purpose of this section is to document everything and cover those fine points that might become a problem for you at some point in the future. Table C.1 covers all the properties that you will configure in the File Transport Properties dialog box. This dialog box is opened when you configure the Address (URI) property in the Receive Location Properties dialog box after selecting the FILE protocol.

**TABLE C.1** File Receive Adapter Configuration Properties

Property Name	Meaning and Settings
Receive folder	Specifies a qualified path to a folder or network share that the receive adapter should poll for messages. This property, a string value, is required.
File mask	A mask describing the filename of messages to retrieve. It may use any of the wildcards permitted by the Windows file system. This property is required and typed as a string. Defaults to *.xml.
Public address Optional.	This property is the public address of the file receive location and is exposed to trading partners; for example, through a SEED package. This property defaults to file://{Receive folder value}/{file mask value}. The literal prefix file:// is required so as to communicate the protocol in use. The property value is a string from 0 to 255 characters.
Retry count	When the receive folder is a network share, this property determines how often the adapter should try to retrieve a file when the share is not immediately available. A long type, this property may take any value from zero (inclusive) to the maximum permitted that type. The default value is five.
Retry interval (min)	The time, in minutes, to wait between retry attempts. This property is a long and may take any value from zero (inclusive) to the maximum permitted for that type. The default is five.
Batch size	This property indicates the maximum number of files the adapter may retrieve and submit to the messaging system in a single batch. A long, this property may take any value from one (inclusive) to the maximum permitted for this type. The default value is 20.

You will generally not need to concern yourself with the Batch size property. It can become a performance issue when you expect to receive large surges of messages. This most often happens in a legacy environment in which the system establishes intermittent communication with a remote application or network and dumps all files that have become available since the last connection in the receive folder. For lower latency, reduce this number because 20 is a value that is best suited to high throughput, taking advantage of parallelism through the architecture.

Additional parameters of the file adapter may be fine-tuned for performance. Unlike the preceding properties, these are controlled through Registry settings. The keys are listed in Table C.2. All are found under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BTSSvs{Guid}\` on the BizTalk server that is hosting the file adapter. If you have multiple servers in the cluster hosting the same receive location for redundancy, be sure to configure each machine. The keys in Table C.2 are not created by default; instead, BizTalk uses the default value given in the table unless you create an entry with a different value. All keys are DWORD values.

**TABLE C.2** Controlling the File Receive Adapter with Registry Settings

Key	Settings
<code>FileReceivePollingInterval</code>	The interval, in milliseconds, that elapses between checking the receive folder for new messages. The default is 60,000 (one minute). The minimum value is 1,000ms (one second), and the maximum is 3,600,000ms (one hour). If the key value is 1, polling is disabled.
<code>RenameReceivedFiles</code>	Enables or disables file renaming when the adapter reads them. The default is 0 (disabled), and the other permissible value is 1.
<code>RemoveReceivedFileRetryCount</code>	The maximum number of times to attempt to delete a file after reading it and submitting it to the messaging system. The default value is 5, the minimum value is 0, and the maximum value is 100.
<code>RemoveReceivedFileMaxInterval</code>	The maximum interval, in milliseconds, the file adapter will wait before trying to delete a processed file. The default is 300,000ms (five minutes), the minimum is 1,000ms (one second), and the maximum is 900,000ms (15 minutes).
<code>RemoveReceivedFileDelay</code>	The initial value for the interval, in milliseconds, that elapses between attempts to delete a processed file. The value doubles after each retry attempt until the maximum value is reached. The default is 10ms, the minimum is 1ms, and the maximum is 1,000ms (one second).

You might wonder about file renaming. Normally, the file adapter will read a file and then delete it when it is successfully submitted to BizTalk. While the pipeline is processing it, it is locked but resident on the file system. When renaming is enabled, the file extension `.BTS-WIP` is appended to the filename after the file is read and before it is submitted to a pipeline. File renaming prevents the creation of multiple messages in the MessageBox when the receive location is shut down while the message is being processed during pipeline processing, and then restarted. When this feature is enabled, the file adapter will ignore files with the extension `.BTS-WIP`.

## Configuring the File Send Adapter

The properties you need to configure for the file adapter when using it to send messages from BizTalk to a disk file are slightly different. These are given in Table C.3.

**TABLE C.3** File Send Adapter Configuration Properties

Property	Meaning and Settings
Destination folder	Specifies the path to the file system folder or network share to which to write the message.
File name	Specifies the name of the file to which the message will be written. Note that the macros from Table C.4 may be used to control the filename.
Copy mode	Defines how the message will be written to a file. The values follow: Append—The message will be appended to the end of the specified file, if it exists, or written into a new file of that name if one does not presently exist. Overwrite—The message will be written to disk even if a file of the same name exists in the target folder. Create new—The default behavior for this property, when selected, causes the file adapter to write the message only if a file of the specified name does not exist. If one does, an error will be generated and send port retry logic will be applied.
Allow cache on write	Defines whether file system caching is permitted when writing a message. This property defaults to false, which ensures maximum reliability but diminishes performance.

You can generate useful filenames when sending a message through the file send adapter through the use of the macros listed in Table C.4. Although many examples use the %MessageID% macro, you can use the various date and time macros or source and destination macros to organize your sent files in more meaningful ways.

**TABLE C.4** File Name Macros Used by the File Adapter

Macro	Meaning
%datetime%	Current UTC date and time in the format YYYY-MM-DDThhmmss.
%datetime_bts2000%	Compatible with the %datetime% macro in BizTalk Server 2000, this macro uses the current UTC date and time expressed in the form YYYYMMDDhhmmss, where sss means seconds (ss) and milliseconds (s).
%datetime.tz%	Local date and time plus time-zone offset in the form YYYY-MM-DDThhmmss(+   -)hmm; for example, 2004-12-24T0530-500 for 5:30 AM in the Eastern Standard Time Zone on 24 December 2004.
%time%	Current UTC time in the format hhmmss.
%time.tz%	Current local time and time-zone offset in the format hhmmss(+   -)hmm.
%DestinationParty%	Name of the destination party—such as the context property BTS.DestinationParty.
%DestinationPartyQualifier%	Qualifier of the destination party—such as BTS.DestinationPartyQualifier.

**TABLE C.4** Continued

%MessageID%	The GUID assigned by the messaging engine to uniquely identify it during processing, it is also the value of the context property <code>BTS.MessageID</code> .
%SourceFileName%	Unqualified name of the file from which the adapter read the message. If the property was not populated because the message was received on another protocol, the macro will not be expanded and will appear as is in the target filename.
%SourceParty%	Name of the source party from which the message was received—such as the context property <code>BTS.SourceParty</code> .
%SourcePartyQualifier%	Qualifier of the source party from which the message was received—such as the context property <code>BTS.SourcePartyQualifier</code> .

**NOTE**

Macros will not be expanded if the underlying property has not been set or if expansion would result in an invalid filename.

## Configuring the File Adapter Programmatically

If you want to programmatically configure a port for the file adapter through the Explorer object model, the .NET class you will need to use is `TransportInfo`, which implements the `ITransportInfo` interface. That interface abstracts the properties that are common to all adapters—essentially, the information you configure in the Port Properties dialog box. The interface does not specify any methods. Programming with this interface is beyond the scope of this appendix, but we do need to shed some light on one topic—how to configure protocol-specific properties.

The solution is found in the `TransportTypeData` property. This read-write property is a string representing an XML property bag. The property bag consists of markup for a root node named `CustomProps` and some child content consisting of elements naming the custom properties peculiar to the protocol in question. The textual content of these elements is the property value. This arrangement allows the specific adapters to share common base class code for the core properties, but still communicate custom properties to the code specific to the transport protocol.

The documentation for each adapter indicates which schema controls the format of the `CustomProps` element for that adapter—for example, `bts_file_properties.xsd` for the file adapter—but these schemas are not included with the product. This is not the problem it might seem, though, because the documentation also lists the custom properties and provides examples of their usage. So long as you remember the `CustomProps` root and wrap each property value in an element named for the property, you will be fine. Each custom property element is an immediate child of `CustomProps`; that is, each such property element is a peer of all the other custom property elements. Table C.5 enumerates the custom properties for the file adapter.

**TABLE C.5** File Adapter Custom Properties by Port Direction

Property Element Name	Values
<b>Receive Properties</b>	
FileNetFailRetryCount	A long value specifying how many times to attempt to access a file share when it is initially unavailable. This property ranges from zero to the maximum size of a long type and defaults to five.
FileNetFailRetryInterval	The interval in minutes that should elapse between attempts to access the receive location on a network share. A long value, this property may range from zero to the maximum size of the long type, but defaults to five minutes.
BatchSize	The number of file-based messages the adapter can submit to the messaging engine in a single batch. Although this is a long type, the range for this property is only 1 to 256, with a default value of 20.
<b>Send Properties</b>	
CopyMode	This long property is analogous to the Copy mode property set through the user interface. The value of this property is enumerated as follows: Append (0), Create new (1), Overwrite (2).
AllowCacheOnWrite	This Boolean property is the equivalent of the Allow Cache on Write property set through the user interface.
ReceivedFileName	This property, typed as <code>xs:string</code> , specifies the full name of the file from which the adapter originally read the message.

## Security and the File Adapter

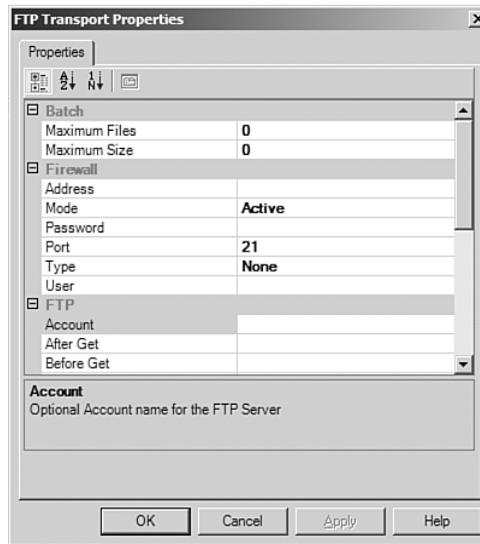
The file adapter needs appropriate rights to the folder where files are received or sent. Because the adapter is enlisted into a specific host, it is the service account for that host that must be granted the rights. For a receive adapter, these rights are read, write, delete files, and delete subfolders and files. The rights are granted for the particular local folder or network share where files will be received. A send adapter, in turn, needs write access to the folder represented by the file URL. In either case, BizTalk Administrator accounts require full access for troubleshooting purposes.

## FTP Adapter

The FTP adapter enables you to send messages to a remote FTP server or receive messages from a remote server via the FTP protocol. In either case, you are working with a remote FTP server. This has caused some confusion in actual use. If you are hosting the FTP server, you either have access to the file system on which the files end up and can therefore use the file adapter, or the FTP server is somehow unavailable (on the other side of a firewall, for instance), in which case it appears as a remote server for the purposes of our discussion. Configuring the FTP adapter, then, is a case of configuring a component with the usual parameters needed for FTP file exchange.

## Configuring an FTP Receive Location

After you select FTP as the transport protocol for a receive location and click the ellipsis button in the Address field to configure it, you will be presented with the FTP Transport Properties dialog box as shown in Figure C.1.



**FIGURE C.1** Configuring an FTP receive location using the FTP Transport Properties dialog box.

The properties you can configure for the FTP adapter in a receive location are listed in Table C.6.

**TABLE C.6** FTP Receive Adapter Properties

Property	Meaning and Settings
<b>Batch</b>	
Maximum Files	Maximum number of files the adapter may submit to BizTalk per batch. A value of zero indicates no limit, and this is the default setting.
Maximum Size	Maximum number of bytes that may be submitted to BizTalk per batch. A value of zero, the default setting, indicates no limit.
<b>Firewall</b>	
Address	URI or IP address of the firewall server.
Mode	Mode to use when connecting to the FTP server. Your choices are Active (default) and Passive.
Password	Firewall password.
Port	Firewall port. This may take a value in the range 1 to 65535, inclusive. The default is 21.
Type	Type of firewall. The choices are None (default), SOCKS 4, and SOCKS 5.
User	Username for the firewall.

**TABLE C.6**    Continued

---

<b>FTP</b>	
Account	FTP account name. Optional.
After Get	FTP commands to issue following the FTP GET. Multiple commands are delimited by semicolons.
Before Get	FTP commands to issue before issuing the GET. Multiple commands are delimited by semicolons.
Error Threshold	Number of FTP errors the adapter can encounter before disabling the receive location. The default is 10.
File Mask	File mask filter specifying which files to retrieve.
Folder	Folder to poll on the FTP server.
Log	Location to which to save the FTP log file.
Max File Size	Permissible maximum size, in megabytes (MB), of a single file for download via FTP. A value of 0 indicates no limit. The default is 100.
Password	FTP account password.
Port	Port number for the FTP server. Defaults to 21.
Representation	File format when downloading. The choices are binary (default) and ASCII.
Server	FTP server name or IP address.
SSO Affiliate	Single sign-on affiliate (if any).
User Name	FTP account username.
<b>Polling</b>	
Interval	Duration of the FTP polling interval. A value of zero tells the adapter to continuously check for files.
Unit	Unit of measure for the Interval property. Valid choices are Days, Hours, Minutes, Seconds.
<b>Tuning Parameters</b>	
Temporary Folder	Location of a temporary folder for use during transfer.

---

SSO affiliates are logical constructs specified by a BizTalk administrator to represent an application, in this case an FTP server, that will participate in BizTalk SSO. The list in the FTP Transport Properties dialog box is initially set to None, but is populated based on whatever affiliate applications have been defined for the BizTalk Server Group.

The use of a temporary folder helps the adapter recover in the event of an FTP transfer failure.

## Configuring an FTP Send Port

Many legacy scenarios involve transferring a file to a trading partner or remote location via FTP. This may be automated in BizTalk using the FTP adapter in conjunction with a send port (see Table C.7).

**TABLE C.7** FTP Send Adapter Properties

Property	Meaning and Settings
<b>Firewall</b>	
Address	Name or IP address of the firewall server.
Mode	Mode in which the adapter will connect to the FTP server. Choices are Active (default) and Passive.
Password	Firewall password
Port	Firewall port (1 through 65535, inclusive). Defaults to 21.
Type	Type of firewall. Choices are None (default), SOCKS 4, and SOCKS 5.
User	Firewall username.
<b>FTP</b>	
Account	Name of the FTP account. Optional.
After Put	FTP commands to issue after the FTP PUT command. Multiple commands are delimited by semicolons.
Allocate Storage	Specifies whether to allocate file storage (required by MVS and AS/400 host systems). Choices are No (default) and Yes.
Before Put	FTP commands to issue before the FTP PUT command. Multiple commands are delimited by semicolons. The OPEN command is not required.
Folder	FTP folder to which to write the message.
Log	Name and location of the FTP log file.
Password	FTP account user password. After it is entered, the value is obfuscated in the dialog box. Clicking the field yields a drop-down edit control.
Port	Port for FTP transfer. The default is 21.
Representation	File format representation during transfer. Choices are binary (default) and ASCII.
Server	Name or IP address of the FTP server.
SSO Affiliate	Name of the SSO affiliate application to use.
Target File Name	Name of the file to write on the FTP server. This name may include the macros specified for the file adapter (see Table C.4).
User Name	FTP account username.
<b>Tuning Parameters</b>	
Temporary Folder	Location of a temporary folder on the FTP server for use during transfer.

**NOTE**

The Temporary Folder property is ignored when connecting to an FTP server on an MVS or AS/400 host. The After Put and Before Put properties are interesting because they enable you to set up FTP server specific information before writing the file, or perform additional commands after writing the file.

## Configuring the FTP Adapter Programmatically

BizTalk enables you to configure the FTP adapter for use with send ports using the Explorer object model. As with the file adapter, the interface to use is `ITransportInfo`.

The procedure is the same: Send custom properties as XML in the `TransportTypeData` property encoded as elements under the `CustomProps` root element. These custom properties are explained in Table C.8.

**TABLE C.8**    Custom Properties for the FTP Adapter

Property Element Name	Meaning
<code>RepresentationType</code>	String denoting how to encode the message for transfer. Choices are binary and ASCII.
<code>SSOAffiliateApplication</code>	Name of the SSO Affiliate application.
<code>UserName</code>	FTP username.
<code>Password</code>	FTP user password.
<code>BeforePut</code>	Commands to issue before the FTP PUT command. Multiple commands are delimited by semicolons.
<code>AfterPut</code>	Commands to issue after the FTP PUT command. Multiple commands are delimited by semicolons.
<code>ReceivedFileName</code>	Full name of the file from which the message was originally read.

## Security and the FTP Adapter

FTP is inherently insecure, and usernames and passwords are sent cleartext. In addition, port information other than the user password is stored in cleartext in SQL Server. Clearly, using the FTP adapter poses risks for sensitive data. Security during transfer is best handled through a dedicated connection or a VPN. Adapter information is protected by limiting access to the server.

## HTTP Adapter

BizTalk's HTTP adapter permits the exchange in either direction of messages with an application or trading partner using the Hypertext Transfer Protocol (HTTP). The adapter is actually composed of two components—one for sending and one for receiving. The receive component is an ISAPI extension (`BTSHTTPReceive.dll`). When setting up a receive location, you will place a copy of this DLL in the folder where you implement your HTTP virtual folder in IIS. An example of an HTTP receive location is given beginning on page 180 in Chapter 6.

### NOTE

`BTSHTTPReceive.dll` is found in the Program Files\Microsoft BizTalk Server 2004\HttpReceive folder for a typical installation of BizTalk. Copy this DLL to the folder in which you want to host your HTTP receive handler.

## Configuring an HTTP Receive Location

The HTTP receive adapter will work with either GET or POST operations. The body of an HTTP POST becomes the body of the incoming BizTalk message, and the query string of

the GET is made into the body of the BizTalk message following that sort of operation. An HTTP code of 202 is returned if the message is successfully submitted. The return code in the case of a failure is a little more complicated. The short answer is that an HTTP code of 500 (internal server error) is returned. The longer answer involves the fact that the adapter submits messages in batches. If a message fails in pipeline processing, the client that submitted that message receives a code of 500, and the other clients with messages in the batch get success codes of 202. If any of the messages fail upon submission to the MessageBox, the entire batch is failed and all clients get a failure code of 500.

Unlike many of the other adapters, the HTTP adapter is hosted by the isolated host because it runs in the IIS process rather than the BizTalk process. By default, the installation program for BizTalk creates an isolated host with the name `BizTalkServerIsolatedHost`. Because IIS is the process that will ultimately host the adapter, you must configure the isolation mode for your receive handler. Windows Server 2003 is the preferred operating system for hosting HTTP receive adapters because IIS 6.0 provides for Worker Process Isolation mode, a more robust and secure mode of hosting IIS applications. In this mode, applications (in our case, the HTTP receive adapter) are grouped into application pools. Each pool forms a process boundary so you can group similar applications on the basis of security privileges and reliability. You can also use IIS 5.0 Compatibility mode, in which case all Web applications share the same process. This is the only mode supported by Windows Server 2000. Because it is limited to 10 simultaneous connections, Windows XP is not recommended for production use and Microsoft does not support it in that role.

#### NOTE

The issues surrounding isolation modes in IIS 6.0 versus IIS 5.0 isolation are discussed at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/iissdk/html/5ccb1199-b55a-42a2-80b1-6522a271a00a.asp>.

After copying the DLL that implements the receive handler, you will need to configure IIS with a virtual directory. If you want to use worker process isolation mode (the default and highly recommended setting), you will also need to configure an application pool. If a Web service extension has not been configured for the HTTP receive adapter, you will also have to add that to IIS's list of extensions.

### Configuring IIS 6.0 for an HTTP Receive Location

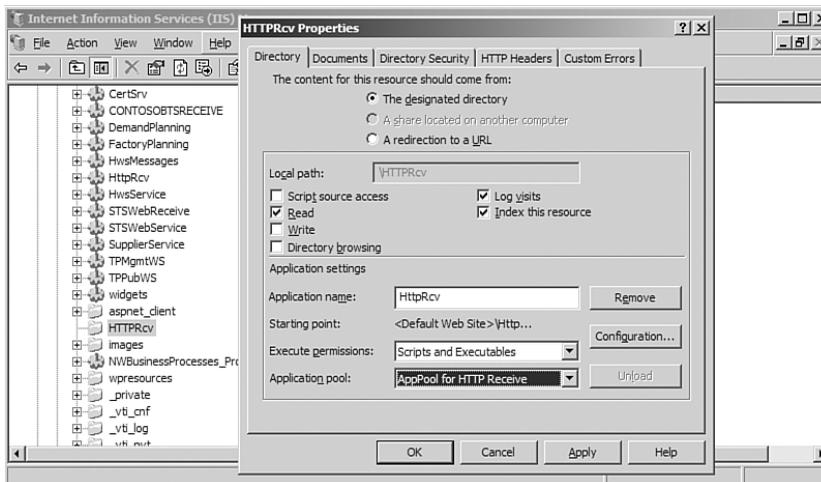
The first thing to do is open IIS Manager and select the Web Service Extensions node in the left pane. If `HwsMessages HTTPReceive` appears in the right pane, the extension was created for Human Workflow Services and you can go on to the next step. Otherwise, you have to make the receive adapter an allowed extension. Click the `Add a New Web Service Extension` link in the right pane of IIS Manager and then type a name for the new extension. This is merely the name that will appear in the list of extensions, so make the name descriptive. Click `Add` and then browse to `BTSHTTPReceive.DLL` in the `HttpReceive` folder under the BizTalk Server installation and check the `Set Extension Status to Allowed` box. IIS will now be able to load the receive adapter for HTTP as needed.

You might have a suitable application pool in existence, but we'll assume you need to create one because this is your first time using the HTTP Receive adapter. Right-click the Application Pools node in the left pane of IIS Manager and select New Application Pool. Type a name for the new application pool into the dialog box and click OK. Now find the new app pool node, right-click, and select Properties to open the Properties dialog box.

Click the Identity page on the Properties dialog box. Check the Configurable option button and enter the name and password of an account that belongs to both the BizTalk Isolated Hosts and IIS\_WPG groups. This gives the app pool access to the BizTalk Credentials database, among other things. Click OK to dismiss the Properties dialog box.

After you've created a folder and dropped a copy of `BTSHTTPReceive.DLL` into it, follow the usual steps to create a virtual directory for your receive location. Be sure to check the box permitting execution of ISAPI applications on the Virtual Directory Access Permissions page of the Virtual Directory Creation Wizard.

Now find the new node for the virtual directory in the left pane and open its Properties dialog box. Be sure that Execute Permissions is set to Scripts and Executables, and then find the application pool you created and select it for the value of the Application Pool property (see Figure C.2).



**FIGURE C.2** You must properly configure the HTTP receive location's virtual directory for execution permissions and application pool when using IIS 6.0 worker process isolation mode.

If you are using IIS 5.0 compatibility mode, or IIS 5.0 under Windows Server 2000, set Execute Permissions to Scripts and Executables and select either High or Medium for the Application Protection property. Then go into the Component Services application (under Administrative tools) and expand COM+ Applications under My Computer on the BizTalk machine. If you selected Medium for Application Protection, right-click IIS Out-of-Process Pooled Applications and select Properties. If you selected High, do the same for `IIS_{Default Web Site//root/virtual directory}`. On the Identity page, select an account that belongs to the BizTalk Isolated Host group.

## Configuring the Receive Location

Now you're finally ready to configure the Receive Location in BizTalk Explorer. Create a receive handler and a receive location under it. The values you can set in the HTTP Transport Properties dialog box are given in Table C.9. The most important is the URL for the virtual directory. Be sure to specify `BTSHTTPReceive.d11` at the end of the URL. The URL may be relative or absolute. Public Address is needed only if you will be exporting information to trading partners. Return Content Type, Loopback, and Return Correlation Handle on Success may be set depending on whether you are setting up a one-way or request-response receive location.

**TABLE C.9** HTTP Receive Location Properties

Property	Meaning and Settings
Virtual directory plus ISAPI extension	URL for the virtual directory plus the <code>BTSHTTPReceive.d11</code> ISAPI extension. This may be a relative URL.
Public Address	Fully qualified URL for the location given in the preceding property.
Return Content Type	Content type of the response returned to the client. This is used only for request-response locations. Defaults to <code>text/xml</code> .
Loopback	Specifies that the received message is either routed to a send port or returned as a response. This is valid only for request-response locations. Defaults to <code>false</code> .
Return correlation handle on success (One-way port only)	If <code>true</code> (default), the receive location will receive the correlation token of the message if it is successfully submitted. This is used only for one-way receive locations.
Use Single Sign On	If <code>true</code> , Enterprise Single Sign-On is used. Defaults to <code>false</code> .

## Configuring the HTTP Adapter for a Send Port

There are two sets of properties to configure for an HTTP send adapter. You can set global properties that will be applied to all instances of the HTTP adapter for a particular host. The send adapter for HTTP is hosted by the normal—that is, nonisolated—host or hosts in BizTalk because HTTP send occurs within the BizTalk Server process rather than proxying through IIS. These properties are configured through the BizTalk Server Administration application by finding the HTTP node under adapters, then finding Send handlers, and then opening the properties dialog box for the host you want to configure. The dialog box has three tabs. The properties you can configure are given in Table C.10.

**TABLE C.10** Global Send Adapter Properties

Property	Meaning and Settings
<b>General Tab</b>	
Host name	Name of the BizTalk host being configured
<b>Properties Tab</b>	
Request timeout (sec)	Duration to wait for a reply from the server. If the value is zero, the adapter will calculate a value based on the size of the message. If no value is provided, the host's timeout will be used.

**TABLE C.10**    Continued

Maximum redirects	Not implemented at this time.
Content type	Content type of the message being sent. Defaults to text/xml.
<b>Proxy Tab</b>	
Use proxy	If true (checked), the adapter will use the server specified in the Server property. Default is false.
Server	Address or name of the proxy server. This property must be set if Use proxy is set to true.
Port	Port number for the proxy server. Defaults to 80.
Username	Username for authentication with the outbound proxy server.
Password	Password for the user account on the outbound proxy server.

Most of the time, you will be configuring a specific send port for use with the HTTP adapter. Many of the properties listed in Table C.11 duplicate the global properties you saw in Table C.10. When you select HTTP as the transport for a send port and click the ellipsis button in the Address (URI) field of the send port's Properties dialog box, you get a Transport Properties dialog box with three tabs.

**TABLE C.11**    HTTP Transport Properties

Property	Meaning and Settings
<b>General Tab</b>	
Destination URL	Address to which to send messages. Include any query strings sent as part of the URL. Maximum length is 256 characters.
Request timeout (sec)	Timeout interval for the HTTP or HTTP/S send. If the interval expires, an error is generated and retry logic for the port is applied.
Maximum redirects	Not implemented at this time.
Content type	Content type of the message being sent. If none is provided, the host's property will be used.
<b>Proxy (Handler Override) Tab</b>	
Use Handler's default	Force use of the proxy server configured on the handler. Defaults to true proxy configuration (checked).
Do not use proxy	Forces the send port to bypass the proxy server.
Use proxy	Forces the send port to use the proxy information provided on this tab.
Server	Proxy server address.
Port	Port to use on the proxy server.
User name	Username for authentication with the proxy server.
Password	Password for authentication with the proxy server.
<b>Authentication Tab</b>	
Authentication Type	Type of authentication to use with the target server. Valid choices are the following: <ul style="list-style-type: none"> <li>• Anonymous (default)</li> <li>• Basic</li> </ul>

**TABLE C.11** Continued

	<ul style="list-style-type: none"> <li>• Digest</li> <li>• Kerberos</li> </ul>
Credentials	Required if Authentication Type is Basic or Digest. Options are the following: <ul style="list-style-type: none"> <li>• Do Not Use Single Sign-On (you will need to provide a user-name and password)</li> <li>• Use Single Sign-On (you will need to select an affiliate application)</li> </ul>
SSL client certificate thumbprint	Thumbprint of the client certificate for use when establishing an SSL connection with the target server.

## Configuring an HTTP Adapter Programmatically

By now you are probably familiar with the routine for filling the `CustomProps` element of the `TransportTypeData` property in the Explorer object model. The element names used to configure a receive location are given in Table C.12, and the same information for a send port using this adapter is given in Table C.13. The restrictions and defaults are as given for their corresponding properties in Table C.9. Table C.12 gives only the property name as used with `TransportTypeData` (that is, the element name), its .NET type, and the name of the property as it appears in BizTalk Explorer.

**TABLE C.12** HTTP Receive Location Adapter Properties for Use with the `TransportTypeData` Property

<code>TransportTypeData</code> Property Name	Type	BizTalk Explorer Property Name
<code>ResponseContentType</code>	string	Return content type
<code>LoopBack</code>	Boolean	Loopback
<code>ReturnCorrelationHandle</code>	Boolean	Return correlation handle on success (One-way port only)
<code>UseSSO</code>	Boolean	Use the BizTalk Server Single Sign On service for user authentication

**TABLE C.13** HTTP Send Port Adapter Properties for Use with the `TransportTypeData` Property

<code>TransportTypeData</code> Property Name	Type	BizTalk Explorer Property Name
<code>RequestTimeout</code>	xs:int	Request timeout (sec)
<code>ContentType</code>	xs:string	Content type
<code>MaxRedirects</code>	xs:int	Maximum redirects
<code>UseHandlerProxySettings</code>	xs:boolean	Use Handler's default proxy configuration
<code>UseProxy</code>	xs:boolean	Do not use proxy/ Use proxy
<code>ProxyName</code>	xs:string	Server
<code>ProxyPort</code>	xs:int	Port
<code>ProxyUsername</code>	xs:string	Username
<code>ProxyPassword</code>	xs:string	Password

**TABLE C.13**    Continued

AuthenticationScheme	xs:string	Authentication Type
Username	xs:string	Username
Password	xs:string	Password
Certificate	xs:string	SSL client certificate thumbprint
UseSSO	xs:boolean	Use Single Sign-On
AffiliateApplicationName	xs:string	Affiliate Application

## SOAP Adapter

This section describes the SOAP adapter supplied in the box. In addition, the WSE 2.0 adapter available for download is built on top of the WSE 2.0 stack, adding WS-Security support. While its configuration is not identical to the SOAP adapter, it is conceptually similar. Most of the time you will be using Web services with orchestration, either by calling a Web service and consuming its return message in your orchestration, or wrapping your orchestration with a Web service proxy. These techniques were illustrated in Chapter 10. Orchestration Designer took care of configuring the SOAP adapter for you in those cases, and that is by far the most common approach to the SOAP adapter. It is, however, entirely possible to use the SOAP adapter without orchestration, and you might need to fine-tune a port configuration generated in conjunction with an orchestration because of specific security needs.

The BizTalk SOAP adapter provides multiple security options. It supports both BizTalk Enterprise Single Sign-On and SharePoint Portal Server Single Sign-On. If the Single Sign-On database is managed by SharePoint, BizTalk will receive the security ticket from that application and map it to the BizTalk security database. SharePoint makes extensive use of Web services itself, and many SharePoint application architectures involve the use of custom Web services. It makes sense to provide support for SharePoint SSO because SharePoint is a natural user interface complement to BizTalk's back-end integration capabilities. Although either form of SSO may be used, only one SSO ticket may be attached to any given SOAP message.

### NOTE

The SOAP adapter uses ASP.NET proxies for receive locations. This makes it subject to all the limitations of .NET Framework XML Serialization. In general, you are safe when working with native XML types for inbound parameters, but you might encounter difficulties when using derived types. Web service calls do not support arrays in this release of the SOAP adapter for receive locations.

## Configuring a SOAP Receive Location

The SOAP adapter, like the HTTP adapter, is hosted by an isolated host instance when you're setting up a receive location. It is advisable to use Windows Server 2003 and IIS 6.0 worker process isolation for the reasons previously given for the HTTP adapter. Note, however, that you cannot host both adapters (HTTP and SOAP) in the same host instance at the same time.

A SOAP receive location is set up when you have an orchestration that you have published as a Web service via the Web Services Publishing Wizard. The wizard creates a proxy Web service implementation using ASP.NET.

You set up a SOAP receive location much like any other receive location in BizTalk Explorer. The properties you can configure in the Transport Properties dialog box after selecting SOAP as the protocol are listed in Table C.14.

**TABLE C.14** SOAP Receive Location Transport Properties

Property	Meaning and Settings
Virtual Directory plus Web Service .asmx file	Relative URL of the Web service implementation as created by the Web Services Publishing Wizard.
Public address	Qualified URL of the Web service implementation. Used when exporting to a trading partner.
Use Single Sign-On	Specifies use of BizTalk Enterprise Single Sign-On.

The receive adapter can distinguish between BizTalk and SharePoint SSO tickets. The underlying adapter code will use the appropriate SharePoint APIs when a SharePoint SSO ticket is received.

Because SOAP is built on top of the HTTP transport in this SOAP implementation, security for a SOAP receive location is handled through IIS Manager. You select the authentication scheme and parameters under the Authentication and Access Control section on the Directory Security tab of the properties dialog box for the .asmx SOAP implementation file. These settings are the familiar HTTP settings you use for any HTTP application or virtual directory.

## Configuring a SOAP Send Adapter

You will use the SOAP send adapter when you want to send a request to a remote Web service and consume the reply. That is, BizTalk will act as a client of the Web service being called. After you have a static send port and select SOAP as the transport protocol, you may configure the properties for the transport as shown in Table C.15. The SOAP Transport Properties dialog box has two tabs.

**TABLE C.15** SOAP Send Adapter Transport Properties

Property	Meaning and Settings
<b>General Tab</b>	
Web Service URL	Address of the Web service being called.
Authentication	Authentication method to use with the Web service. Choices are <ul style="list-style-type: none"> <li>• Anonymous (default)</li> <li>• Basic</li> <li>• Digest</li> <li>• NTLM (Kerberos)</li> </ul>
Credentials	Available only for Basic or Digest type authentication. Choices are

**TABLE C.15** Continued

	<ul style="list-style-type: none"> <li>• Do Not Use Single Sign-On (you will need to provide a username and password)</li> <li>• Use Single Sign-On (you will need to select an affiliate application)</li> </ul>
Client Certificate Thumbprint	Thumbprint of the client certificate to use when connecting to a secure Web service.
<b>Proxy Tab</b>	
Use Handler's default proxy configuration	Forces the send port to use the configuration of the host handler (defaults to true).
Do not use proxy	Override the handler and do not use the proxy configuration.
Use proxy	Override the handler and use the proxy configuration provided.
Server	Name of the proxy server to use.
Port	Name of the port to use with the proxy server (defaults to 80).
User name	Proxy server user account name.
Password	Proxy server user account password.

## Configuring the SOAP Adapter Programmatically

Table C.16 provides the association between the element names to use when configuring a SOAP receive location programmatically and the equivalent property names in BizTalk Explorer.

**TABLE C.16** SOAP Receive Location Custom Properties

TransportTypeData	Property Name	Type	BizTalk Explorer Property Name
URI		xs:string	Virtual Directory plus Web Service .asmx file
AddressableURI		xs:string	Public address
UseSSO		xs:boolean	Use Single Sign-On

The equivalent information for a send port using the SOAP send adapter is given in Table C.17.

**TABLE C.17** SOAP Send Adapter Custom Properties

TransportTypeData	Property Name	Type	BizTalk Explorer Property Name
URI		xs:string	Web Service URL
Username		xs:string	Credentials (user name)
Password		xs:string	Credentials (password)
ClientCertificate		xs:string	Client Certificate Thumbprint
AffiliateApplicationName		xs:string	Credentials (affiliate application name)
UseProxy		xs:boolean	Do not use proxy/Use proxy
ProxyAddress		xs:string	Server
ProxyPort		xs:int	Port
ProxyUsername		xs:string	Username
ProxyPassword		xs:string	Password

## What About Web Services Enhancements (WSE)?

If you've followed the development of Web services, you know that a number of follow-on extensions are proposed for use on top of SOAP. These are collectively known as WS-X; the X represents Security, Policy, Addressing, or one of the other Web service extensions. WS-Security, in particular, provides for a very robust signing and encryption regime in which the author can specify the algorithm to use, the type of certificate, and which portion of the message is signed or encrypted. None of this is built into Microsoft BizTalk Server 2004 and the SOAP adapter for the simple reason that the extensions were not finalized in time.

Microsoft does provide an implementation of some of these standards, notably WS-Security, in the Web Service Enhancements (WSE) 2.0 package of .NET classes. WSE 2.0 is available at <http://www.microsoft.com/downloads/details.aspx?FamilyId=FC5F06C5-821F-41D3-A4FE-6C7B56423841&displaylang=en>. These capabilities are due to be supported in the next version of Windows Server under the SOA messaging system currently code named Indigo.

The BizTalk product team is well aware of this, of course. The team released a WSE-capable adapter in beta format in November 2004. This adapter is now a released adapter available for free download at <http://www.microsoft.com/downloads/details.aspx?FamilyID=ba643360-e9dc-4fc5-8d60-8e6c5eca7861&displaylang=en>.

## SMTP Adapter

The SMTP adapter uses an SMTP server, such as the SMTP service provided with Windows Server, to transmit messages via email. The SMTP adapter provides only a send capability. If you wish to receive messages via email, you will need to do some custom programming using a server such as Exchange or obtain an unsupported POP3 adapter from [gotdotnet.com](http://gotdotnet.com). A future version of BizTalk Server will provide support for POP3 in the box. The most common use of the SMTP adapter is to send some sort of notification to a human user as a result of some processing in an orchestration. An excellent example is to send an InfoPath document to a user in response to some processing that requires human intervention. The form can then use the File or Web services protocols to send a reply back to BizTalk.

The SMTP adapter sends the message as an email message attachment. You can configure the adapter to request delivery and read receipt notification to a specified email URL. Contrary to some of the documentation and whitepapers surrounding BizTalk, a custom pipeline is not required if you are sending a basic XML or flat-file message. Indeed, re-serializing an XML stream into XML with an XML send pipeline is both redundant and decreases performance. A custom pipeline is required, however, when you are sending InfoPath forms. InfoPath is triggered by the presence of some processing instructions in the XML document. To add these in BizTalk, you will need to configure the XML Assembler pipeline component. For most messages, however, all you will need to do is configure the adapter as discussed in the following section.

## Configuring the SMTP Adapter

The BizTalk Explorer Transport Properties dialog box for the SMTP adapter has two tabs. The properties you can configure through this dialog box are given in Table C.18.

**TABLE C.18**    SMTP Adapter Properties

Property	Meaning and Settings
<b>General Tab</b>	
To	Email URL to which to send the message. You may specify more than one URL using a separator supported by the SMTP server.
CC	This property must be set. The maximum length is 256 characters. Email URL to which to send copies of the message. You may specify more than one URL. Maximum length is 1,024 characters.
Subject	Subject line sent with every instance of the message. The maximum length is 256 characters.
Notification	Type of notification desired. The notification is sent to the URL specified for the From property on the Handler Override Tab. Choices are <ul style="list-style-type: none"> <li>• Read Receipt</li> <li>• Delivery Receipt</li> </ul>
<b>Handler Override Tab</b>	
SMTP server name	Machine name of the SMTP server.
From (e-mail address)	Email URL to place on the STMP From line.
Authentication type	Type of authentication to use with the SMTP server. Choices are <ul style="list-style-type: none"> <li>• (Default)</li> <li>• No authentication</li> <li>• Basic authentication</li> <li>• Process account (NTLM)</li> </ul>
User name	Account name to use when authenticating with the SMTP server.
Password	Password to use when authenticating with the SMTP server.

Most of the macros defined in Table C.4 may be used when forming the Subject, To, From, and CC properties. The macros that may be used are %MessageID%, %datetime\_bts2000%, %datetime%, %datetime.tz%, %time%, %time.tz%, %SourceParty%, %SourcePartyQualifier%, %DestinationParty%, and %DestinationPartyQualifier%. If the expanded macro would result in an invalid STP property or the underlying system property has not been set, the macro will not be expanded.

## Configuring the SMTP Adapter Programmatically

Like the other adapters, the SMTP adapter may be configured using the `ITransportInfo` interface's `TransportTypeData` property. The element names used as children of the `CustomProps` root element in that property value are specified in Table C.19, along with their data type and the corresponding name in the BizTalk Explorer user interface.

**TABLE C.19**    SMTP Adapter Custom Properties

TransportTypeData	Property Name	Type	BizTalk Explorer Property Name
SMTPHost		xs:string	SMTP server name
From		xs:string	From (email address)

**TABLE C.19** Continued

CC	xs:string	CC
Subject	xs:string	Subject
SMTPAuthenticate	xs:int	Authentication type (valid choices are 0 for no authentication, 1 for basic authentication, and 2 for NTLM authentication)
SendUserName	xs:string	Username
SendPassword	xs:string	Password
ReadReceipt	xs:Boolean	Notification (Read Receipt)
DeliveryReceipt	xs:Boolean	Notification (Delivery Receipt)

## Sample SMTP Configuration

You seldom see an SMTP example, so let's set up a simple one right now. The idea is to be able to drop an XML message in a file receive location, have it taken up by BizTalk messaging, and send it via SMTP to a known mailbox.

### NOTE

The file `binding.xml` supplied with the code download for this appendix may be used with the BizTalk Deployment Wizard's Import BizTalk Assembly Binding from File option to import the settings described in the following section. Be sure to inspect the configuration and modify such things as SMTP hostname and To and From URIs to conform to your local environment.

The first thing to do is create a message schema. The content of the message is not important. In fact, if you want to take a shortcut, you can use any schema you have deployed on your BizTalk Server. Assuming you have no suitable schemas deployed, go ahead and create a schema for an instance message like the one you see in Listing C.1. That message has a document element named `SMTPRoot`, with a child field element of `Message` whose only purpose is to hold a text message for use in our example.

### TIP

Don't worry about the prefix or the namespace. The message in Listing C.1 was developed by creating the schema with a default generated namespace and then using the Generate Instance option. The content of the message is unimportant to our example.

### LISTING C.1 Sample Test Message

```
<ns0:SMTPRoot xmlns:ns0="http://SchemaForSMTP.DummySchema">
  <Message>What did you expect?</Message>
</ns0:SMTPRoot>
```

Don't forget to create a keyfile and set the Assembly Key File property for the project. After you have done this, build and deploy the project using Visual Studio .NET.

Next, create a receive port. Give it the name **MsgRcv**. Click OK to accept the defaults. Now create a receive location for the port. Name it **MsgRcvLocation**. Select FILE as the Transport Type property value, BizTalkServerApplication (or its equivalent on your system) as the Receive Handler, and the default XMLReceive pipeline as the Receive Pipeline. Click the ellipsis button in the field for the Address (URI) property to open the Transport Properties dialog box.

Enter a local file folder as the receive folder. I'm using C:\temp\MsgRcv on my system. Enter \*.xml as the file mask and click OK. Click OK in the Receive Location Properties dialog box. Use BizTalk Explorer to enable the file receive location.

So far we haven't done anything related to SMTP. That's about to change. After we have an XML message in the messaging system, we want a send port with a subscription to our file receive port to take the message and send it as an XML attachment to an email. Perform the following steps to configure your SMTP send port:

1. Right-click Send Ports in BizTalk Explorer and select Add Send Port. Accept Static One-way Port. Click OK.
2. In the Port Properties dialog box, enter the name **SMTPSend** as the port name.
3. Select SMTP from the list of adapters under the Transport Type Property.
4. Click Address (URI) to enter the Transport Properties dialog box.
5. On the General tab, enter the URL of a mailbox on your system; for example, jsmith@mycorp.com, in the To field.
6. Enter **Sample XML Message** in the Subject field.
7. On the Handler tab, enter the name of your SMTP server. If your BizTalk server has the SMTP service installed (check IIS Manager for this), for example, you may enter the machine name of your BizTalk Server.
8. Enter a valid email URL for the From property.
9. If your SMTP service has special security requirements, configure the Authentication Type property and, if necessary, the Username and Password.
10. Click OK to complete the Transport Properties dialog box.
11. Click the Send node in the left pane of the Send Port Properties dialog box. Select Microsoft.BizTalk.DefaultPipeline.XMLTransmit for the pipeline.
12. Click the Filters & Maps node, and then the Filters subnode in the left pane. Click the empty line to configure a new filter expression.
13. Select BTS.ReceivePortName as the Property, == as the Operator, and then enter **MsgRcv** as the Value. Recall that this is what we named the receive port.
14. Click OK to complete the send port configuration.
15. Right-click the Send Port node in BizTalk Explorer and select Start to enlist and start the send port.

Now that you have your ports configured, you can drop an XML message conforming to your schema into the folder for the receive location. If you have done everything correctly, you can open Outlook or some other email client for the mailbox you used in step 5 and see a new message with the subject Sample XML Message bearing your XML message as an attachment.

## MSMQT Adapter

BizTalk Server supports MSMQ through a Web downloadable MSMQ adapter. In general, customers should use the MSMQ adapter first. MSMQT was designed to support in-order messaging and is single threaded. This design trade-off means that it supports in-order messaging well but only scales through additional receive locations rather than performance of a single adapter. In contrast, the MSMQ adapter (which also supports large files) provides multithreaded scale. You can use this when performance is paramount and message order is not important. For example, if your workflow involved sending or receiving MSMQ messages with no related follow-on traffic, as when sending notices of action taken, the downloadable MSMQ adapter would be your best choice. This adapter is BizTalk's link to message queuing on the Windows platform. The adapter breaks with the messaging adapters in previous versions in that it does not use MSMQ itself for its operation. Instead, BizTalk implements a queuing system that is compatible with the MSMQ 1.0 wire format, interoperates with MSMQ, yet guarantees reliable, in-order delivery of messages to servers in a BizTalk cluster. It is this guarantee that required the break with MSMQ. MSMQ can be configured for reliability, and messages in the queue are in order of delivery, but if several servers in a cluster were to poll a single queue and experience a fault in operation, reliability might be lost, and message sequencing certainly would be lost. Consequently, BizTalk uses the MessageBox database to emulate a queuing system. Because it controls the database, it can control which server gets which message and detect when a problem occurs in the cluster.

There are no actual queues that you can browse in MSMQ; rather, you provide a name when setting up a receive location and BizTalk treats it as a virtual queue. When sending messages, BizTalk transmits the message to either a real MSMQ queue or a virtual MSMQT queue on another BizTalk server using the MSMQ wire format.

The MSMQT send adapter works with direct format names. It will send to both private and public MSMQ queues when Active Directory is installed. Receive locations appear to MSMQ as private queues.

### Registering the Adapter

Although the MSMQT adapter is provided with the installation of BizTalk Server 2004, the component is not registered during installation for much the same reason that Windows Server 2003 does not automatically enable HTTP services. Before using the adapter, therefore, you must register the adapter with BizTalk.

You will perform this task with the BizTalk Administration snap-in to MMC. After opening it, find the Adapters node under the BizTalk Server node, right-click the node, and select New Adapter. When the Add Adapter dialog box opens, enter a descriptive name and select

MSMQT from the drop-down list under Adapter. Click OK to close the dialog box. The adapter will now appear in the list of available adapters for the server group.

## Configuring the MSMQT Receive Adapter

Setting up a receive location for MSMQT is relatively straightforward. There is only one tab to the Transport Properties dialog box. You have to provide a name for the queue you want to configure and provide some additional information. The properties for an MSMQT receive location are given in Table C.20.

**TABLE C.20**    MSMQT Receive Adapter Properties

Property	Meaning and Settings
Queue name	Name of the queue to monitor.
Requires MSMQ Authentication	When checked, the queue will reject any message that does not use MSMQ protocol-level authentication. Unchecked by default.
Non-Transactional Queue	When checked, the queue will accept only nontransactional messages. Unchecked by default.
Queue number	A number used to identify the queue when using MSMQ private format addressing. Any value greater than five is acceptable. Although this field defaults to six, it is highly recommended that you change this value.

Because all MSMQT queues are virtual (in terms of being emulated in the MessageBox), the Queue name property is largely arbitrary in most cases. Good practice would indicate using some name that reflects the use of the queue. There are two important restrictions, however, if you want to interoperate with IBM MQSeries queues via the Host Integration Server bridge. If you want to receive a message from MQSeries, the queue name must be less than 25 characters, and the name must be all uppercase.

The Queue number property comes into play if you want to use private addressing from MSMQ. To do this, Active Directory must be installed. The BizTalk server hosting the queue will be identified by a GUID rather than a name, and this GUID is discovered from the directory. The format for such addressing is `PRIVATE={server GUID}\{queue number}`.

### TIP

The BizTalk documentation as of the December 2004 refresh, incidentally, states that Queue number must be greater than or equal to 4, with 1 through 3 reserved. The Transport Properties dialog box itself, however, states that the number must be greater than 5 and defaults the value to 6. It is a good idea to change that default and establish a range or private number of your own—for example, 5985—that you will commonly use. This is a practice sockets programmers have used for years.

## Configuring the MSMQT Send Adapter

Configuring a send port for the MSMQT adapter is also an easy task. The dialog box has one tab and two properties. These properties are explained in Table C.21.

**TABLE C.21** MSMQT Send Adapter Properties

Property	Meaning and Settings
Destination Queue	MSMQ or MSMQT address of the destination. DIRECT format is recommended, but PUBLIC and PRIVATE are supported if Active Directory is installed.
Use MSMQ Authentication	When checked, MSMQ protocol-level authentication will be used when a message is sent on this port.

**NOTE**

Contrary to the BizTalk Server documentation as of the December 2004 refresh, the MSMQT adapter will send messages to public queues, provided Active Directory is installed. DIRECT format addressing is recommended, and PUBLIC and PRIVATE addressing is generally reserved for use with the MQSeries bridge or cases where special routing considerations apply.

## Base EDI Adapter

The Base EDI Adapter for BizTalk supports the X12 and EDIFACT EDI standards and allows organizations using BizTalk to interoperate with trading partners using EDI. EDI specifies its own communications protocols and message-handling procedures, so a special Base EDI service is needed to perform these functions. The EDI Adapter uses the file system as a point of contact with the rest of BizTalk. A file can be dropped into a configured folder and the adapter will pick it up and send it to the Base EDI service as if the service were receiving it via EDI reception at an organization's EDI port. Similarly, the EDI adapter can listen at a particular EDI port for a message leaving the Base EDI service through the EDI protocol, grab it, and drop it to a file location.

**NOTE**

The Base EDI adapter is not particularly strong in terms of EDI functionality and for most EDI customers the Covast EDI Accelerator will be a better fit.

**NOTE**

When BizTalk Server 2004 is installed, the Base EDI Service is set to Disabled. Before working with the EDI Adapter, you must start this service through the Services option of Administrative Tools.

## Configuring the EDI Adapter

If you look at the EDI node of the Adapters folder in the BizTalk Server Administration console, you will see that the EDI Adapter is preconfigured with one receive handler and one send handler. You may add additional handlers to suit the needs of your application. The Administration console is the only place you can manually configure handlers. You will use BizTalk Explorer to set up send and receive ports and receive locations.

The Base EDI Receive Handler receives messages from EDI and drops them to a file location. When you add an EDI Receive Handler, the Receive Handler Wizard guides you through the steps to configure the receive handler for EDI. The pages in the wizard are the same as the property pages you see when you edit an existing EDI Receive Handler. The dialog box for this is shown in Figure C.3.

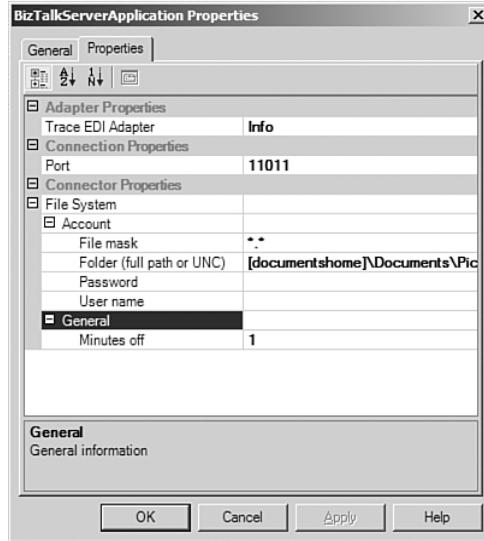


FIGURE C.3 Configuring properties for an EDI Receive Handler.

The properties for EDI Receive Handlers are given in Table C.22.

TABLE C.22 EDI Receive Handler Properties

Property	Meaning and Settings
<b>General Tab</b>	
Host	Select the BizTalk host application for the receive handler. By default, this is BizTalkServerApplication.
<b>Properties Tab</b>	
Trace EDI Adapter	Sets the level of logging to the event log. This property must be set. You may select from the following settings: <ul style="list-style-type: none"> <li>• Error</li> <li>• Info</li> <li>• Off</li> <li>• Verbose</li> <li>• Warning</li> </ul>
Port	The IP port number used to receive messages from the EDI service. This property must be set.

**TABLE C.22** Continued

File	Mask The filename mask for messages that will be written to disk after getting them from EDI. This property is mandatory. At runtime, the asterisk wildcard (*) will be replaced with an interchange number. The default setting is *.edi.
Folder	This optional property specifies the folder where messages received from EDI will be written.
Password	Password of the account needed for accessing a network share. Optional.
User Name	Account name for accessing a network share when writing EDI messages to disk. Optional.
Minutes off	Number of minutes the file system connector is inactive between writing EDI interchanges to disk. Optional.

The send handler takes messages from BizTalk and forwards them via the EDI protocol. Configuring a send handler for the EDI adapter works the same way as configuring a receive handler. The properties for EDI send handlers are given in Table C.23.

**TABLE C.23** EDI Send Handler Properties

Property	Meaning and Settings
<b>General Tab</b>	
Host name	Specifies the BizTalk host associated with the send handler. Defaults to BizTalkServerApplication.
<b>Properties Tab</b>	
Logical Address	This mandatory property specifies the address to which to send messages. This address follows the conventions of the EDI protocol.
Trace EDI Adapter	Sets the level of logging to the event log. This property must be set. You may select from the following settings: <ul style="list-style-type: none"> <li>• Error</li> <li>• Info</li> <li>• Off</li> <li>• Verbose</li> <li>• Warning</li> </ul>
Port	IP port used to send messages. Mandatory.

## Configuring a Receive Location for EDI

After you have send and receive handlers, you can create send and receive ports using the EDI protocol. As is always the case with native adapters, the configuration details that are specific to EDI are configured through the Transport Properties dialog box accessed when you click the ellipsis button in the Address (URI) field on the receive location's properties dialog box. The settings for an EDI receive location are given in Table C.24. The Transport Properties dialog box has one tab divided into four sections.

**TABLE C.24** Configuration Properties for an EDI Protocol Receive Location

Property	Meaning and Settings
<b>Adapter</b>	
Logical Address	Specify the EDI address where the message will be received. This is selected by choosing from a list of URIs you establish when you create an EDI alias for a party.
<b>Flags</b>	
Functional Acknowledgements	Specify whether to send EDI acknowledgements. Choices are <ul style="list-style-type: none"> <li>• Always</li> <li>• Never</li> </ul>
<b>EDIFACT Documents</b>	
(document)	Select the document to use at this location by expanding categories and selecting messages and envelopes from the nodes contained therein.
<b>X12 Documents</b>	
(document)	Select the document to use at this location by expanding categories and selecting messages and envelopes from the nodes contained therein.

## Configuring a Send Port for EDI

A send port takes a message from the BizTalk messaging system via the file connector and gives it to the EDI Base Service for transmission using the EDI communications protocol. When configuring a send port for use with EDI, then, you need to provide information regarding the EDI destination and the file location from whence the message will be taken up. In addition, you have the opportunity to configure EDI-specific items. The properties for an EDI protocol send port are given in Table C.25.

**TABLE C.25** EDI Send Port Properties

Property	Meaning and Settings
<b>Adapter</b>	
Logical Address	The EDI protocol URI to which the message will be sent. This is associated with a party alias, as was the case for receive locations.
<b>Connector</b>	
Connector	Select the connector to use to get the message from BizTalk for sending via EDI. At present, File System is the only choice.
File mask	Specifies the filename mask of files to poll for in the file system folder. The wildcards * and ? may be used.
Folder	Full local file path or UNC name of a network share to poll for messages arriving from BizTalk.
Password	Password to use when accessing network shares
User name	Account name to use when accessing network shares.

TABLE C.25 Continued

Flags	
Error level	Specifies the level at which an error causes a failure. Choices are <ul style="list-style-type: none"> <li>• Interchange—Failure of a message fails the entire interchange.</li> <li>• Message—Failure of a messages fails only that message; the remainder of the interchange might be successful.</li> </ul>
Functional acknowledgements	Specify when to send an EDI acknowledgement. Choices are <ul style="list-style-type: none"> <li>• Always</li> <li>• Never</li> </ul>
Options	
Component separator	Specifies the character used to separate segments of a composite element. The EDI specifications provide this information.
Decimal notation	Specifies the decimal point character for the message. Each EDI document format has default settings that this property overrides. Any settings made at the trading partner level, in turn, override this value.
Element separator	Specify the character used to separate singular (noncomposite) elements.
Release character	Sets the escape character when a reserved character must be used in a message. Not used in X12.
Segment tag separator	Character used to separate segment tags.
Segment terminator	Character used to terminate a message segment.
Wrap segment	Indicates whether segments that are too long should be wrapped and how that wrapping occurs. Choices are <ul style="list-style-type: none"> <li>• No</li> <li>• Yes (CRLF)</li> <li>• Yes (LF)</li> </ul>
Supported Document Types for EDIFACT	
(document)	Select the EDIFACT document allowed for use with this port from the list.
Supported Document Types for X12	
(document)	Select the X12 document type allowed for use with this port from the list.

## SQL Server Adapter

The SQL adapter allows BizTalk to send data to or retrieve data from a Microsoft SQL Server database using the XML updategram scheme supported by Microsoft SQL Server. An **updategram** is an XML format that represents a SQL SELECT, UPDATE, or INSERT command. The SQL adapter relies on the SQLXML capability that has been available for SQL Server for some time now. That capability provides a layer that translates between resultsets and XML documents and communicates via HTTP. The SQL adapter makes use of this by communicating with SQL Server using SOAP.

The SQL receive adapter issues a SQL command at a specified polling interval. The results are brought into BizTalk as an XML message. The SQL Server send adapter allows BizTalk to send an updategram message to the specified database, resulting in persistent changes in the data contained therein.

## Installation Prerequisites

The SQL Server Adapter has two dependencies that must be installed prior to installing BizTalk Server. The first is SQLXML 3.0 SP 2. This is the add-in that allows SQL Server to accept and create updategrams in XML. If this is not installed, you will be prompted to install it during installation of BizTalk Server.

The adapter uses the COM-based SOAP Toolkit 2.0 to communicate with SQL Server. If it is not installed before BizTalk, attempts to use the SQL Adapter will fail.

## Configuring a SQL Receive Location

The hardest part of working with the SQL adapter is the message schema. This is because you will seldom want to deal with the updategram schema itself. The easiest way to get a SQL updategram schema for use with the SQL adapter is to generate an item for your BizTalk solution. From the Solution Explorer, right-click your project and select Add, Add Generated Items. When the Generated Items dialog box appears, select the Add Adapter template and click Open. You will be presented with a list of registered adapters. Select SQL. The SQL Transport Schema Generation Wizard will lead you through a series of steps in which you indicate which server and database you want to use, the connection string, and other information typically used when connecting to a SQL Server database. One item is the SQL command or stored procedure to execute. When you complete the wizard, an updategram schema and a blank orchestration configured to use the updategram in a request-response port are created and added to your solution. You may refer to the schema when configuring receive locations. SQL Receive Location properties are explained in Table C.26.

**TABLE C.26**    SQL Receive Location Properties

Property	Meaning and Settings
<b>Receive Configuration</b>	
Poll While Data Found	If set to true, the adapter will continue to submit batches for INSERT or UPDATE queries or stored procedures until the SQL command fails to return data. If false, only a single effort will be made per polling interval. Note: Do not set this property to true for SELECT statements or stored procedures that are guaranteed to return data because an endless loop might result. Defaults to false.
Polling Interval	Interval between attempts to poll SQL Server. Defaults to 30.
Polling Unit of Measure	Unit of measure for the polling interval property. The choices are <ul style="list-style-type: none"> <li>• Hours</li> <li>• Minutes</li> <li>• Seconds (default)</li> </ul>
<b>SQL Configuration</b>	
Connection String	Enter the database connection string or click the ellipsis button to generate one from a dialog box that elicits standard database connection information—for example, security mode, initial database, and so on.
Document Root Element Name	Document root element name for the updategram.
Document Target Namespace	Target namespace for the updategram.

**TABLE C.26** Continued

SQL Command	Enter a SQL command or click the ellipsis button to enter a reference to a generated schema. The Import information from a generated schema dialog box is presented, and you will have the opportunity to refer to BizTalk projects and schemas for the updategram.
URI	A URI identifying the receive location. The format is SQL://server_name/database_name[additional_string], where <i>server_name</i> is the name of the database server (or . for the local machine) and <i>database_name</i> is the name of the database with which you want to communicate. The additional string is available to make a unique URI in cases where your solution uses two or more receive locations that employ the same database. BizTalk requires unique URIs for each location.

**NOTE**

If you use the ellipsis button in the SQL Command property field to access the Import information from a generated schema dialog box and refer to a generated schema, the Document Root Element Name, Document Target Namespace, and SQL Command properties will be populated for you.

## Configuring a SQL Send Port

Now that you have some experience with SQL adapter ports, you'll find that configuring a SQL send port is slightly easier than configuring a receive location. The SQL Transport Properties dialog box has one tab with one section, Adapter Properties. Those properties are given in Table C.27.

**TABLE C.27** SQL Send Adapter Properties

Property	Meaning and Settings
Connection String	Enter a connection string or click the ellipsis button to access the Data Link Properties dialog box and enter database configuration properties.
Document Target Namespace	Enter the namespace of the outgoing updategram.
Response Root Element Name	Enter the document element of the outgoing updategram.

Notice that a transport properties dialog box for an SMTP send port will not let you reference a generated schema as we were able to do for a SQL receive location. You can generate this information and then inspect the generated schema to get the information you need.

**TIP**

The BizTalk native adapters are the components you will use time and again to connect applications using the messaging system. Although the list of properties available for configuration might seem daunting, they are manageable when you realize that they are tied directly to the transport protocol they represent.

The procedure used to create ports is the same for all the native adapters. What differs is the transport-specific configuration for the receive locations and send ports.

