



**JNBridge JMS Adapter for BizTalk Server
User's Guide
Version 3.0**

www.jnbridge.com

JNBridge, LLC
www.jnbridge.com

COPYRIGHT © 2007-2011 JNBridge, LLC. All rights reserved.

JNBridge is a registered trademark and JNBridgePro and the JNBridge logo are trademarks of JNBridge, LLC.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows Vista, Windows 7, Windows Server, BizTalk and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other marks are the property of their respective owners.

October 26, 2011

Contents

How to use this guide	5
Overview of the JNBridge JMS Adapter for BizTalk Server	6
How it works.....	6
Installing, Licensing and Configuring the JMS Adapter	8
Supported Platforms for JNBridge JMS Adapter for BizTalk Server	8
Target Machine Prerequisites	8
Installing the JMS Adapter	8
Licensing.....	10
License files.....	10
Licensing and application configuration files	10
Evaluation licenses.....	11
On-line activation.....	12
Off-line activation.....	12
License managers	13
64-bit vs. 32-bit installation and configuration.....	14
32-bit platforms.....	14
64-bit platforms.....	14
BizTalk Configuration of the JMS Adapter	15
Adding the JMS Adapter to BizTalk Server	15
Configuring Send and Receive Adapter Handler Properties.....	16
JMS Properties Category	17
JNBridge Properties Category.....	18
Security Properties Category	19
Batch Properties Category	20
Debugging Properties.....	20
Removing or Nullifying Property Values.....	20
Configuring Send Ports and Receive Locations	21
Creating and Configuring Send Ports	21
Adapter Behavior Category.....	22
Connection Properties Category	23

USERS' GUIDE

- JMS Operation Properties Category 23
- Override Transmit Handler Properties Category 25
- Creating and Configuring Receive Ports and Locations 25
 - Adapter Behavior Category 25
 - Connection Properties Category 26
 - JMS Operation Properties Category 26
 - Override Transmit Handler Properties Category 28
 - Receive Polling Properties 28
- Messaging Patterns 29
 - Request Response receive ports 29
 - Solicit Response send ports 29
- Using Message Properties and Map Messages 30
 - Properties, XML Schema and .NET namespaces 30
 - JMS Receive Properties 30
 - JMS Send Properties 32
 - JMS Configure Properties 34
 - Using Map Messages 36
- Trouble Shooting 37
 - Deploy JMS Header Schema to BizTalk Application 37

How to use this guide

This guide contains information about installing, configuring, and using the JNBridge JMS Adapter for BizTalk Server. It is organized according to the particular task or tasks you wish to perform with the JMS Adapter.

- Please read the section, *Installing, Licensing and Configuring the JMS Adapter*, before attempting to install and configure the adapter.
- All users are encouraged to read the section, *Overview of the JNBridge JMS Adapter for BizTalk Server*. It gives information on the architecture of the JMS Adapter, how it works and the various scenarios in which it can be used.
- Users who have purchased licenses or are planning a deployment of the JNBridge JMS Adapter for BizTalk Server, should read the section, *Licensing the JNBridge JMS Adapter*, which describes the various licensing mechanisms, and discusses the various ways of deploying license keys.

Overview of the JNBridge JMS Adapter for BizTalk Server

The JNBridge JMS Adapter for BizTalk Server is a BizTalk Adapter Framework implementation that encapsulates the Java Message Service client capabilities into a standard BizTalk adapter. Within BizTalk Server, the adapter can be configured as endpoints bound to receive and send ports allowing a BizTalk Orchestration to consume and produce messages to and from JMS queues and topics.

The JMS Adapter uses interoperability technology from JNBridge to integrate any vendor's JMS 1.1 compliant implementation into BizTalk Server. This includes OpenJMS, JBoss, IBM WebSphere MQ, Tibco Enterprise Message Service, BEA WebLogic, Oracle Enterprise Message Service, Pramati JMS Server, SonicMQ and Sun ONE Message Queue.

The JNBridge JMS Adapter exposes a simple interface for configuring send and receive handler and location properties. The JMS Adapter requires no modification to the JMS server. Nor does it require any vendor specific native support for the Windows platform—all that is required is a vendor's standard Java implementation. The support for any vendor's JMS implementation is accomplished through the JNBridge Java and .NET interoperability product, JNBridgePro. For more information on JNBridgePro, visit www.jnbridge.com.

How it works

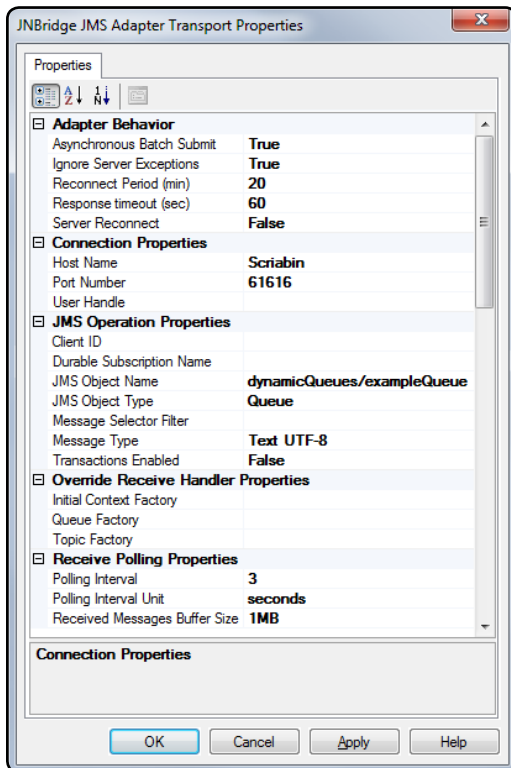


Figure 1. A receive location showing the JMS Adapter configuration

The JNBridge JMS Adapter integrates any Java Message Service (JMS) client into BizTalk Server.

As part of the deployment of a BizTalk Server Orchestration, the adapter can be configured as endpoints bound to send and receive ports, allowing the BizTalk orchestration to produce and consume messages to and from JMS queues and topics. The binding configures the ports to connect to specific JMS queues and topics on a specific JMS server, shown in Figure 1.

Once the configuration and binding is completed, the orchestration is started. Any document directed to a send port bound to the adapter will be sent to the corresponding JMS queue or topic as a JMS message. Any JMS message appearing on a queue or topic bound to a receive port will be received from that queue or topic and enter the orchestration as a document at that receive port.

The underlying technology that supports any JMS implementation is JNBridgePro, JNBridge's Java and .NET interoperability product. JNBridgePro can be used to access any Java API from .NET, or any .NET API from Java. The

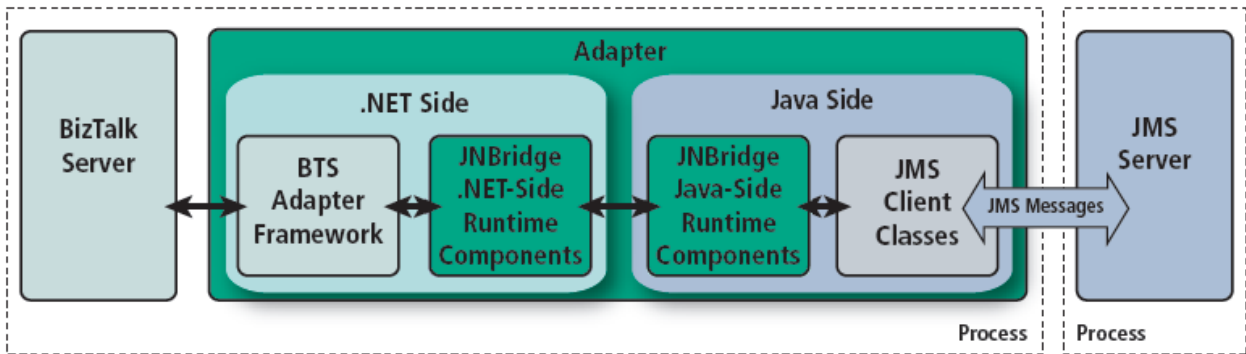


Figure 2. Architecture of the JNBridge JMS Adapter for BizTalk Server

JMS adapters implement JNBridgePro’s interoperability technology to access the JMS API automatically for you. Figure 2 shows the architecture of the JNBridge JMS Adapter for BizTalk Server.

The adapter contains both a .NET side running in a CLR and a Java side running in a JVM, where the CLR and the JVM are running in the same process. JNBridgePro manages the inter-platform communications. The JMS client classes are those provided by the vendor, which guarantees that the adapter will work with any vendor’s JMS server. The adapter utilizes the BizTalk Adapter Framework to communicate with the BizTalk Server.

Installing, Licensing and Configuring the JMS Adapter

The JNBridge JMS Adapter for BizTalk can be installed and configured on any machine running BizTalk Server 2006, BizTalk Server 2006 R2, BizTalk Server 2009 and BizTalk Server 2010.

Supported Platforms for JNBridge JMS Adapter for BizTalk Server

- Microsoft Windows XP and XP Professional. Both x86 and x64 platforms supported.
- Windows Server 2003. Both x86 and x64 platforms supported.
- Windows Vista. Both x86 and x64 platforms supported.
- Windows 7. Both x86 and x64 platforms supported.
- Windows Server 2008. Both x86 and x64 platforms supported.
- Windows Server 2008 R2 x64.

Target Machine Prerequisites

The following prerequisites are needed for either the x86 or x64 platforms.

- A Java Run-time Environment (JRE) must be installed on the target machine. The JNBridge JMS Adapter supports the Standard Edition JRE 5 or above. The JRE may be download from java.sun.com/javase/downloads. For 64-bit platforms, if the x64 version of the adapter has been installed, a x64 JRE must be installed.
- For any given JMS implementation, the JMS Java client environment must be installed including all JAR files as well as any additional configuration such as environment variables or J2EE configuration files.
- Microsoft BizTalk Server 2006, 2006 R2, 2009 and 2010.

Installing the JMS Adapter

The installation of the JMS Adapter requires administrator privileges on the target machine. The installation consists of a single installation.

Start the installation by executing the Windows Installer, e.g. `InstallBizTalkJMSAdapter_300.msi` for a 32-bit platform. For a 64-bit platform, use `InstallBizTalkJMSAdapter_300_x64.msi`. This is a simple installation allowing the user to accept the EULA (End User License Agreement) and to choose a target installation directory.

If updating from version greater than or equal to 1.2, it is not necessary to uninstall the previous version before installing version 3.0.0. If updating a version previous to 1.2, it is necessary to uninstall that version. If updating, all BizTalk host instances must be shut down.

The following runtime components are installed:

- The assemblies JNBridgeBTS2006JMSAdapter.dll, and JNBridgeBTS2006JMSProperties.dll, are placed in the adapter's installation \bin directory. The second assembly is also placed in the GAC.
- The assemblies, JNBShare.dll, JNBShareMem.dll, JNBridgeJMSCore.dll and jnbproxies.dll, provide the .NET-to-Java bridge between the Java Virtual Machine and the Common Language Runtime. They are installed in the GAC.
- The library, rlmXXX_x86.dll or rlmXXX_x64.dll, contains licensing functionality and is placed in the \bin directory.
- The assembly Microsoft.Samples.BizTalk.Adapter.Common.dll, is part of the BizTalk Adapter Framework and is installed in the \bin directory.

The assembly, JNBridgeBTS2006JMSAdapterManagement.dll, provides configuration capabilities. It is placed in the \bin directory.

Troubleshooting installation exceptions

- If updating from version 1.2 to 3.0, it is not necessary to uninstall version 1.2. Any version prior to 1.2 will have to be uninstalled. However, uninstalling a previous version does not require removing the adapter from BizTalk (though all host instances must be shut down).
- If updating to 2.1 from a previous version, the new request/response and solicit/response capabilities must be enabled. This can be done by removing the adapter from BizTalk and then re-adding it back in. However, this is a real pain as all send and receive ports configured for the adapter will have to be removed. A simpler solution is to update the BizTalk Management Database, BizTalkMgmtDb, using this SQL statement:

```
UPDATE adm_Adapter SET Capabilities=15755 WHERE  
MgmtCLSID='A72F4E84-CFF0-4CF6-AC49-1E995BD36AAD'
```

- ! ***The x86 version of the adapter will install and work on a 64-bit platform in a 32-bit host instance using a x86 JRE. The x64 version of the adapter will not install or work on a 32-bit platform. The x64 version must be installed on a 64-bit platform and use a x64 JRE.***
- ! ***It is possible to manually install the JRE by simply copying the folder containing the engine to someplace on the target machine's file system. A manual installation of the JRE will not place the Java VM in the Windows system folder, \Windows\system32, a requirement for the JNBridge JMS Adapter. Simply copy java.exe from the JRE bin directory to the system directory.***

USERS' GUIDE

Licensing

The JMS adapter must be licensed on each machine on which it is installed. JNBridge supports a variety of license types. The licensing types are similar to JNBridgePro. For more information, please visit <http://www.jnbridge.com/jnbpro-licensing.htm>

! *If you have used versions of the JMS adapter older than version 3.0, note that starting with v3.0 we have introduced a new licensing mechanism. The concepts and components described below will be different from those you might have used in previous versions.*

License files

For most types of licenses, the license is encapsulated in a license file, which is a text file whose suffix is *.lic*. (The file's name is generally assigned by JNBridge's license tracking mechanism.) When a license file is present, the adapter will look for the license file in the adapter's root installation directory, e.g. `C:\Program Files\JNBridge\JMSAdapters\BTS2006`.

If an invalid license file is found, an *InvalidLicenseException* is thrown. The adapter will register the exception in the system application event log. If error logging is enabled in the adapter, the *InvalidLicenseException* will also be logged. A license file can be invalid for a number of reasons, including:

- It has been tampered with.
- It is time-limited and has expired.
- It is node-locked, and is being used on a machine other than the one to which the original license was locked.

Licensing and application configuration files

You have the option of specifying the location of your license file, or, if you are using a license server, the location (host and port) of the license server. This can be done by modifying the configuration file for the BizTalk host process, `BTSNTSvc.exe.config` or `BTSNTSvc64.exe.config`.

Inside the `<configuration>` section of the BizTalk configuration file, add the following section if it is not already there:

```
<configSections>
  <sectionGroup name="jnbridge">
    <section name="licenseLocation" type="System.Configuration.SingleTagSectionHandler"/>
  </sectionGroup>
</configSections>
```

Inside the <jnbridge> section, add the following line:

```
<licenseLocation directory="absolute/relative path to directory containing license file"/>
```

or

```
<licenseLocation host="license server machine" port="license server port"/>
```

Choose the first variant if you are using a license file; choose the second variant if you are using a license server. Again, specifying this information is optional. If you omit it, the licensing mechanism will continue looking for the license in the adapter's root installation directory.

Evaluation licenses

As part of the download process, you will be emailed an activation key for a 30-day evaluation license. When you receive this key, enter it in the JMS adapter's registration tool, shown in Figure 3, under the Online License Activation tab and click on the Activate License button. This will allow you to use the JMS adapter on a trial basis for 30 days. (See the section, *On-line activation*, below, for additional details.) You must perform this action before you can begin your evaluation.

The evaluation license file that is obtained through this activation is node-locked, and cannot be copied to other machines. If do you wish to place evaluation installations of the adapter on additional BizTalk machines, you can enter the same activation key into the registration tool on the new machines at any time and receive 30-day evaluation licenses on those machines. Please note that any deployments and installations using evaluation licenses obtained through this activation key will stop working after the 30-day evaluation period expires. Also note that, under the terms of your JNBridge JMS adapter for BizTalk Server license, you may not use your evaluation license to run production applications.

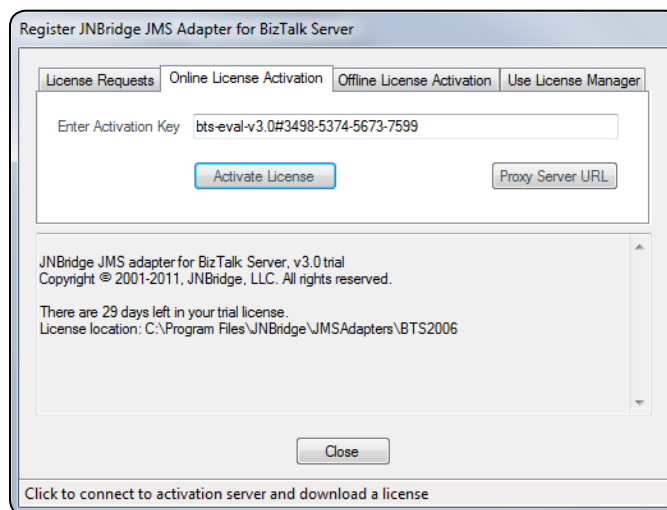


Figure 3. The Registration Tool and online activation

USERS' GUIDE

It is important to note that this activation key cannot be used twice on any given machine. If you need an extension to your 30-day evaluation period, please visit <http://www.jnbridge.com/sweval.htm> and follow the instructions there, or contact registration@jnbridge.com.

In the event that your deployment does not have Internet connectivity, please follow the instructions in the section “Off-line activation,” below.

On-line activation

Once you have purchased adapter licenses, you will receive an activation key. If the machines on which you will be licensing the adapters have access to the Internet, you can use this activation key to claim your licenses. To do so, you must use the adapter registration tool (RegistrationTool.exe) on each of the machines to be licensed. The registration tool will be part of the installation. Launch the registration tool, and select the Online License Activation tab (Figure 3).

Paste the activation key into the Enter Activation Key slot and click on the Activate License button. If you have licenses left to activate, the operation will succeed and your installation will be licensed. The license file will be placed in the adapter's root installation directory.

If you have no more licenses, or the operation otherwise fails, you will see an error message indicating the reason for the failure.

If application deployment and activation are part of an automated process, you can also perform the activation operation as part of a script. In this case, run the registration tool as follows:

```
RegistrationTool.exe /key=activation_key /licpath=path_to_folder_that_will_contain_license_file
```

If you leave out the /licpath option, the license file will be written to the folder containing RegistrationTool.exe.

If you believe you have Internet access, an attempt to activate a license may result in this error message:

This error may be the result of using a proxy server for internet access. If a proxy server is being used, then online activation requires that the environment variable HTTP_PROXY be set before running the Registration Tool. The syntax for setting the HTTP_PROXY environment variable is: HTTP_PROXY=http://[proxy_server_url]:[port]/

Alternatively, you may click on the button, Proxy Server URL, on the Online License Activation tab and set the proxy server's hostname and port number. If you are using a script to activate licensing, then the above mentioned environment variables can be set in the script using the `setlocal` command

Off-line activation

In the event that the machines on which you want to license the adapter do not have Internet connectivity, you can claim a license by launching the registration tool, then selecting the Offline License Activation tab (Figure 4a).

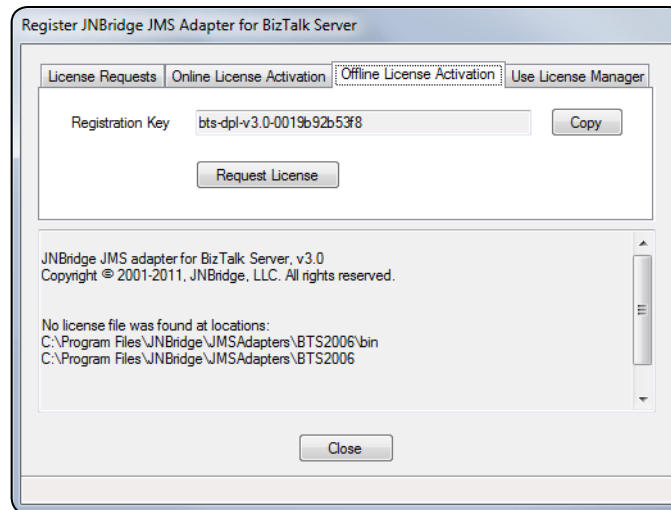


Figure 4a. Using off-line activation

Click the **Request License** button and follow the instructions in the displayed page, or copy the displayed registration key into an email and mail the license request to registration@jnbridge.com, or visit <http://www.jnbridge.com/swreg.htm> and follow the instructions there, supplying the registration key when requested. In response, you will be emailed a license file that you should deploy to the adapter root installation directory, e.g. `C:\Program Files\JNBridge\JMSAdapters\BTS2006`.

License managers

Certain types of licenses, including floating licenses, as well as licenses that will run on terminal servers and virtual machines, require the use of a license manager. If you have purchased these licenses, JNBridge will supply you with a license manager, which you should install and configure according to instructions that are included with the license manager.

To use a license served up by a license manager, you will need to configure the adapter to point to the license manager. See *Licensing and application configuration files*, above, for a discussion of how to do this.

The registration tool provides some assistance in doing this. When the **Use License Manager** tab is selected (Figure 4b), the user has the option of entering the hostname and port of the license manager (that is, the hostname of the machine on which the license manager resides, and the port on which it is listening). Once those values are entered, clicking on the **Load License** button will test whether these values are correctly configured, and, if they are, clicking on the **Export Config File** button will cause a fragment of XML containing the configuration information to be output to a file. The contents of the file can be incorporated into the application configuration file.

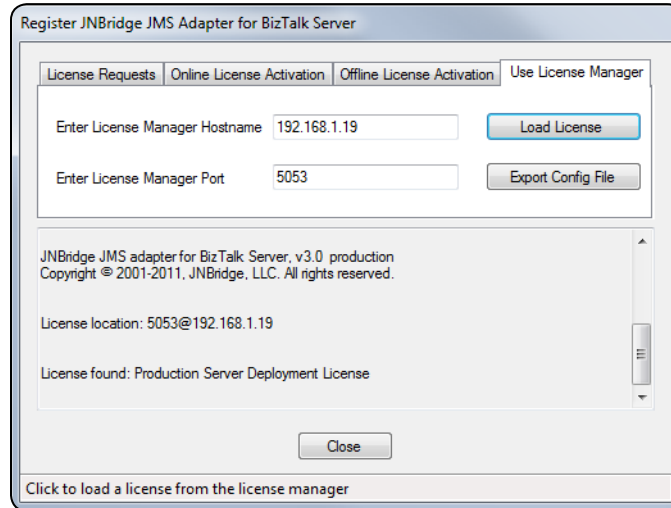


Figure 4b. Pointing to a license manager

Please note the following:

- Clicking the Load License button does not obtain a license for the application being licensed. It is still necessary to add information to the application configuration file that points to the license manager.
- If you have clicked on the Load License button, we recommend that you exit the registration tool before running your application.
- Use of the functionality in the Use License Manager tab is completely optional and is provided as a convenience. It is entirely possible to configure the application manually, without the assistance of this tool.

64-bit vs. 32-bit installation and configuration

The JNBridge JMS Adapter for .NET comes in two versions: A x86 version to target 32-bit platforms and a x64 version targeting 64-bit platforms.

32-bit platforms

Only the x86 version of the adapter can be installed on a 32-bit platform. The x64 version will not install nor will it work on a 32-bit platform. It is possible to install the x86 version of the adapter on a 64-bit machine, however a 32-bit BizTalk host instance must be used.

■ Required JRE

The x86 version of the adapter requires a 32-bit JRE. Do not install a 64-bit JRE on a 32-bit machine.

64-bit platforms

The x64 version of the adapter can only be installed on a 64-bit platform. A 64-bit BizTalk host instance must be used

■ Required JRE

A x64 platform JRE is required. The x64 version of the adapter will not work with a x86 JRE.

BizTalk Configuration of the JMS Adapter

After the JNBridge JMS Adapter for BizTalk Server has been installed, it must be added to BizTalk and configured using the BizTalk Server Administration Console, shown in Figure 5.

Adding the JMS Adapter to BizTalk Server

In the BizTalk Server Administration Console find the Adapters folder under the Platform Settings folder and right-click to bring up the context menu. Choose the New ► Adapter... menu choice to bring up the Adapter Properties dialog box shown in Figure 5. Using the Adapter Properties dialog box, click on the drop-down control Adapter: and choose JMS. The user may also enter a more meaningful name and description for the adapter. Click on OK. The JNBridge JMS Adapter will now reside in the list of adapters in the Adapters pane of the BizTalk Administration Console.

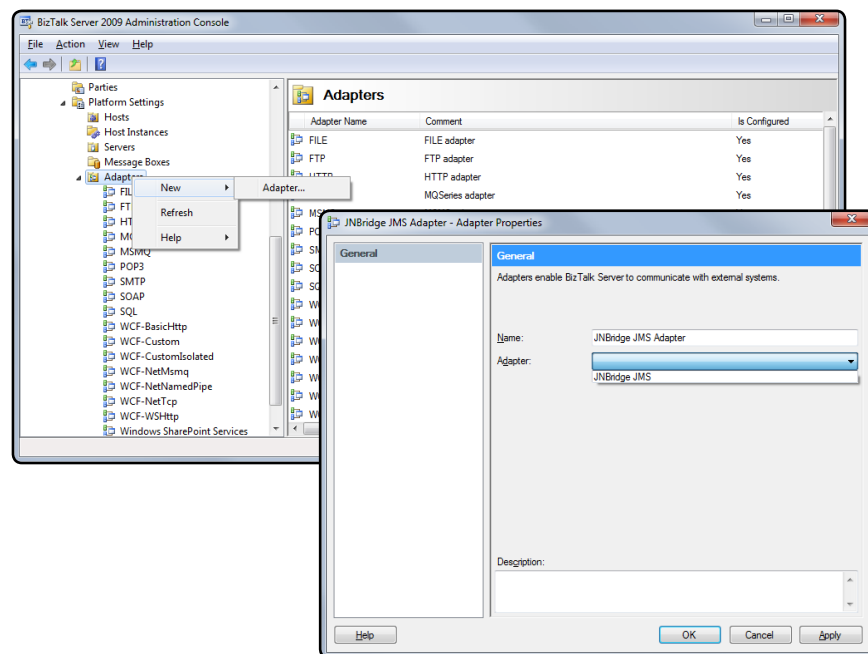


Figure 5. Adding the JNBridge JMS Adapter to BizTalk

Configuring Send and Receive Adapter Handler Properties

In the BizTalk Administration Console, double click the JMS Adapter in the Adapters pane to open up the Send and Receive properties, shown in Figure 6. Double click either Send or Receive to open the Adapter Handler Properties dialog box. Click on the Properties button to open the Transport Properties grid, also shown in Figure 6. This grid contains handler transport properties global to all send or receive ports configured to use the JNBridge JMS Adapter. In other words, all JMS Adapter send or receive ports will inherit these transport properties. You must configure send handler transport properties in order to produce messages to queues and topics. Likewise, you must configure receive handler transport properties in order to consume messages from queues and topics. In most cases, the values of the properties will be identical between the send and receive handlers; however, depending on the JMS server implementation, they may be different.

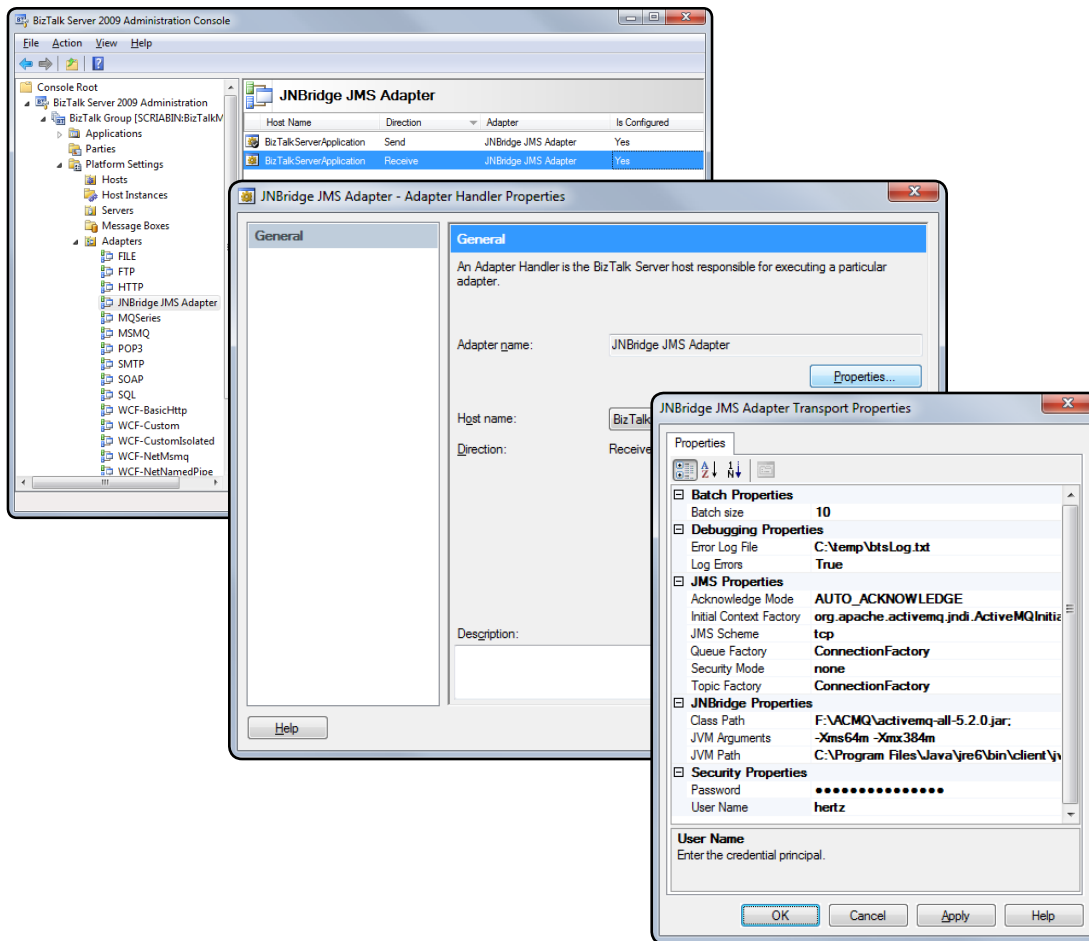


Figure 6. Accessing the Handler Transport Properties dialog

JMS Properties Category

In the Adapter Transport Properties dialog, Figure 6, the JMS Properties category are properties used to properly connect to a JMS server. In most cases, you should be able to obtain the correct property values from the administrator of your JMS server.

■ JMS Acknowledge Mode

The Acknowledge Mode is a drop-down list containing the JMS specification that determines how a JMS client and server institute a reliable messaging protocol. The choices are AUTO_ACKNOWLEDGE, CLIENT_ACKNOWLEDGE and DUPS_OK_ACKNOWLEDGE. Regardless of the choice, the JNBridge JMS Adapter will correctly implement the protocol. For most JMS implementations, AUTO_ACKNOWLEDGE is the default configuration.

■ Initial Context Factory

This is a text-editable field containing the name of the initial context factory. The initial context factory is a Java class that provides instances of various JMS and JNDI classes required for a complete JMS client. While the JNBridge JMS Adapter encapsulates the implementation details of a JMS client providing a simple abstraction of the message service, it is still necessary to choose among the various initial context factories configured for a particular JMS implementation. The initial context factory shown in the Transport Properties grid in Figure 6 is the default initial context factory for ActiveMQ.

! *Factory names are case sensitive—be sure the name (including the complete namespace, if necessary) is typed correctly.*

■ JMS Scheme

This is a text-editable field. The JMS Scheme, or protocol, is particular to each vendor's JMS implementation and the types of connections supported. Each vendor may support several schemes such as http, https or rmi (remote method invocation). The scheme name does not have to reflect a protocol—the scheme for Tibco EMS is tibjmsnaming, for example.

■ JMS Security Mode

The JMS Security Mode is a drop-down list that specifies the type of security required by the JMS server implementation. The choices are none and simple. If the choice is simple, then the server expects a user name and password.

! *If the JMS server implements simple security, it is not necessary to configure this property. Enter a user name and password—the JNBridge JMS Adapter will automatically switch to the simple security mode.*

USERS' GUIDE

■ Queue Factory

This is a text-editable field. The queue factory produces instances of classes used to connect to queues hosted on the JMS server. Each JMS vendor may have several queue factories implementing different properties.

■ Topic Factory


This is a text-editable field. The topic factory produces instances of classes used to connect to topics hosted on the JMS server. Each JMS vendor may have several topic factories implementing different properties.

JNBridge Properties Category

The JNBridge Properties Category correctly configure the .NET-to-Java interoperability core components.

■ Class Path

The Java class path is a set of semicolon-separated paths to the JAR or class files required for a JMS client installation. The class path is used by the JNBridge Java and .NET interoperability components to locate the JMS and JNDI client Java classes so they can be instantiated in the Java Virtual Machine.

To edit the class path, click in the field to enable the browse button . Click on the button to launch the Edit Class Path dialog, Figure 7. The class path can be rearranged by selecting a class path element and clicking on the Move Up or Move Down buttons, or by selecting an element and clicking on the Delete button. The files and folders in the system CLASSPATH environment variable can be added to the class path list by clicking on the Add system CLASSPATH button. Note that only checked elements will be added to the Java class path when the dialog is dismissed.

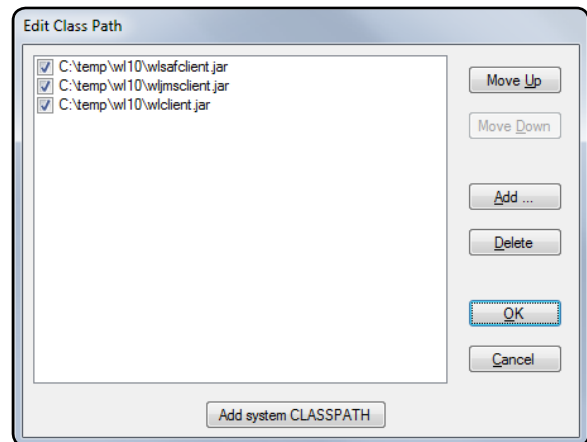


Figure 7. The ClassPath editor

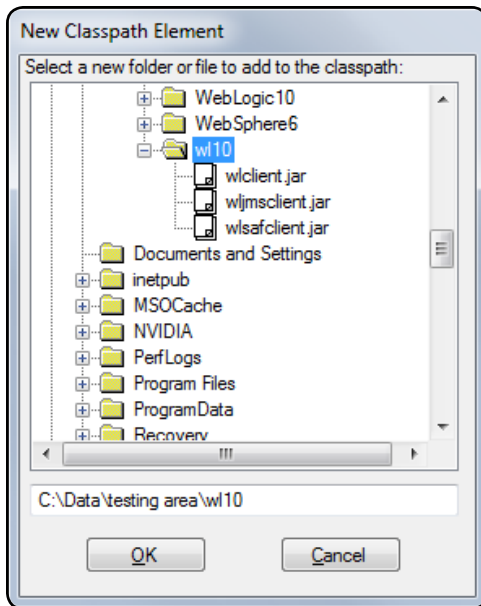


Figure 8. The New Classpath Element dialog


To add a folder or JAR file to the class path, click on the Add... button. This causes a New Classpath Element dialog, shown in Figure 8, to be displayed. In this dialog, the user can navigate to the desired folders or jar files, or can enter a file path directly.

The New Classpath Element dialog supports multiple selection—multiple folders and/or JAR files may be selected by ctrl-clicking, while a range of folders and/or JAR files may be selected by shift-clicking. Clicking on the OK button will cause the indicated folders or files to be added to the Edit Classpath dialog.

■ JVM Arguments

This property is used to send arguments to the Java Virtual Machine. In Figure 6, the arguments shown set the JVM object heap minimum and maximum sizes. Other arguments can be used to set system properties, etc.

■ JVM Path

The JVM Path property is the absolute path to the Java Virtual Machine implementation, `jvm.dll`. To edit the JVM Path property, click in the field to enable the browse button.  Click on the button to launch the standard File Open dialog. Navigate to `jvm.dll` and click OK.

Security Properties Category

This category provides security credentials necessary to connect to a JMS server, if the JMS implementation supports security mode `simple`. If the JMS implementation supports security mode `none`, then security credentials are not required.

■ Password

Click in this field to drop-down the password edit field. Type in the password.

■ User Name

This is a text editable field. Enter the user name necessary to connect to the JMS server.

USERS' GUIDE

Batch Properties Category

This category allows you to optimize the number of messages in a batch. This allows the JNBridge JMS Adapter to process messages in groups rather than one at a time. Setting a batch value will diminish overhead associated with processing messages.


- Batch size

This is a text-editable field. Enter the size of a batch.

Debugging Properties

This category provides means to log errors to a user selected file.

- Error Log File

Choose a error log file. Use the browse button  to open the Open File dialog. Within the dialog, use the context menu to create and rename files.

- Log Errors

Toggle this from False to True to disable and enable error logging to the configured file.

Removing or Nullifying Property Values

It may be necessary to remove or nullify property vales that have been set in the Transport Properties grid. If you try and highlight the value and strike the Delete or Backspace key, the error message shown in Figure 9 will appear.

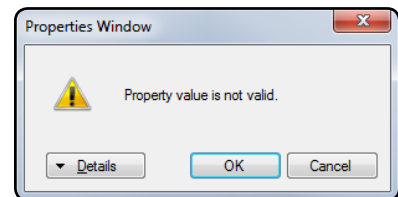


Figure 9. An annoying error

In order to correctly null a value, you must place the cursor over the property name, e.g. Initial Context Factory, and right-click to obtain a context menu that allows you to null the property field, shown in Figure 10.

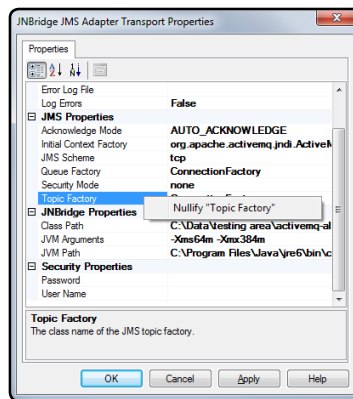


Figure 10. Using the context menu to null property values

Configuring Send Ports and Receive Locations

Figure 11 shows a simple BizTalk orchestration that receives messages from a queue and forwards them on to a topic. Once this orchestration is deployed into BizTalk Server, it is necessary to bind the logical ports ReceiveQueue and PublishTopic to actual send and receive ports created and configured using the BizTalk Server Administration Console. The JNBridge JMS Adapter for BizTalk supports static send and receive ports, request/response receive ports, solicit/response send ports and dynamic send ports.

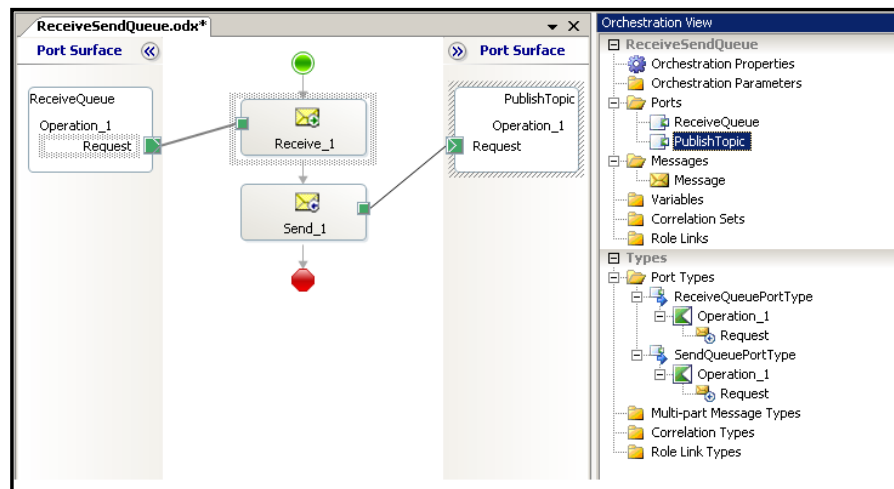


Figure 11. A simple orchestration

Creating and Configuring Send Ports

Figure 12 shows the creation of a Static One-way Send Port by right-clicking in the Send Ports pane of a deployed application and choosing New. This will bring up the Send Port Properties dialog shown in Figure 13 on the next page. For information on using dynamic send ports, see the section *Using Dynamic Send Ports*. Solicit/Response port properties use the same properties shown in Figure 13, next page. For more information on Solicit/Response send ports, see the section *Messaging Patterns*.

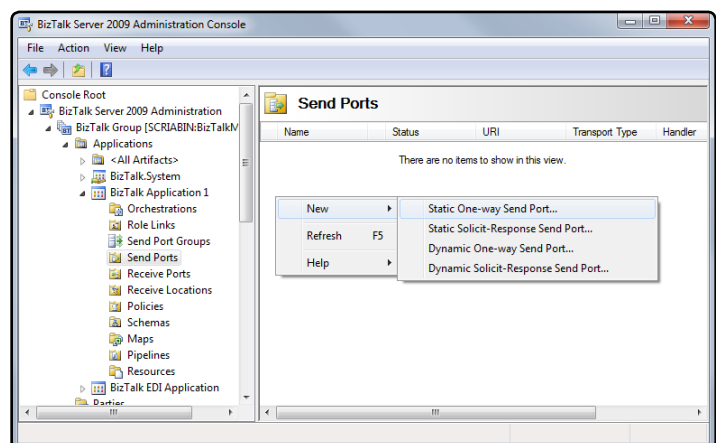


Figure 12. Creating a Static One-way Send Port

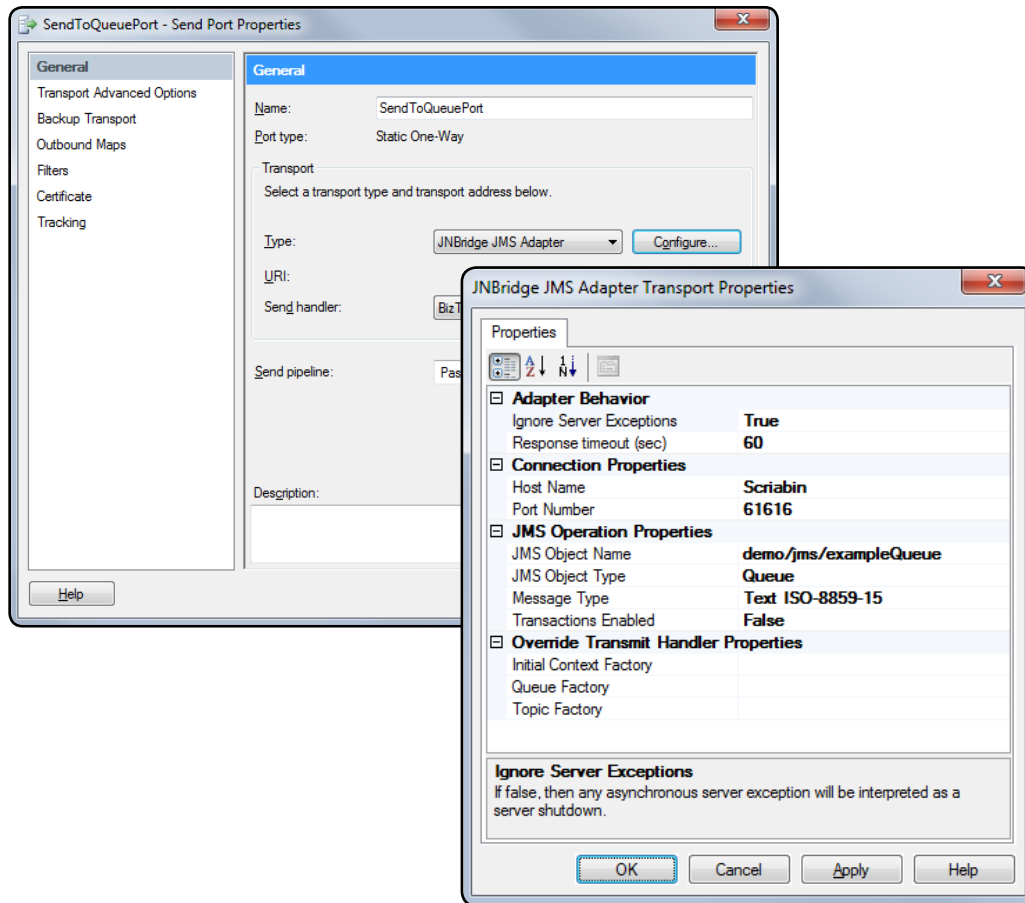


Figure 13. Send Port Properties and the Send Transport Properties dialogs

In the Send Port Properties dialog, click the drop-down control Type: and choose the JNBridge JMS Adapter. Click the Configure button to bring up the Transport Properties dialog, Figure 13, above. This property grid contains the location properties that determines JMS server location, message type and whether to transmit to queues or topics.

Adapter Behavior Category

The Adapter Behavior category, shown in Figure 13, contains two properties.

■ Ignore Server Exceptions

If a subscribed message is available for a send port, the adapter will first check to see if the JMS server has sent any asynchronous exceptions. If the property Ignore Server Exceptions is True, then any asynchronous exceptions sent by the JMS server will be ignored. If False, then the send port will disconnect if *any* asynchronous exception is sent by the JMS server.

The send location will then post an error event and suspend the message. When the next subscribed message is available for the send port, the adapter will attempt to reconnect to the JMS server. If the server is still unavailable, the adapter will post an error event and suspend the message. The send port will not disable if the server is unavailable, but will continue to attempt a re-connection with each subscribed message. If the adapter is able to reconnect to the server, an event is posted and the message is transmitted. All suspended messages can then be transmitted.

The JMS specification does not define an exception class for server shutdown, it only specifies that the server must send an asynchronous exception upon shutdown. This is why any asynchronous exception is considered a server shutdown if Ignore Server Exceptions is False.

- **Response time-out**

This property is used only for Solicit/Response send ports. It is the period of time a solicitation will wait for a response. If a time-out occurs, an exception will be thrown and the solicitation message will be suspended.

Connection Properties Category

Figure 13, preceding page, shows the category Connection Properties. These properties determine where the JMS server resides and the port number where it is listening for connections.

- **Host Name**

This a text-editable field. Enter the name or IP address of the machine hosting the JMS server.

- **Port Number**

This is a text-editable field. Enter the port number on which the JMS server is accepting connections.

JMS Operation Properties Category

These properties determine what operation the send port will enable.

- **JMS Object Name**

This is a text-editable field. Enter the JNDI name bound to the JMS queue or topic. Depending on the requirements of the JMS implementation, it may be necessary to supply a complete path, e.g. `java:comp/resource/jms/myQueue`.

- **JMS Object Type**

This is a drop-down list containing two values: Queue or Topic.

USERS' GUIDE

■ Message Type

This is a drop-down list containing several values: Text, Text UTF, Text ISO-8859-15 or Bytes. If any of the Text types are chosen, then the JNBridge JMS Adapter will send a JMS Text Message converting the binary encoding correctly. If Bytes is chosen, then the JNBridge JMS Adapter will send a JMS Bytes Message.

■ Transactions

Enable or disable local JMS transactions. When transactions are enabled, any resulting error during transmission will result in the message being removed from the JMS queue or topic. The send port will throw an exception and the message will be retransmitted dependent on retry properties set by the user.

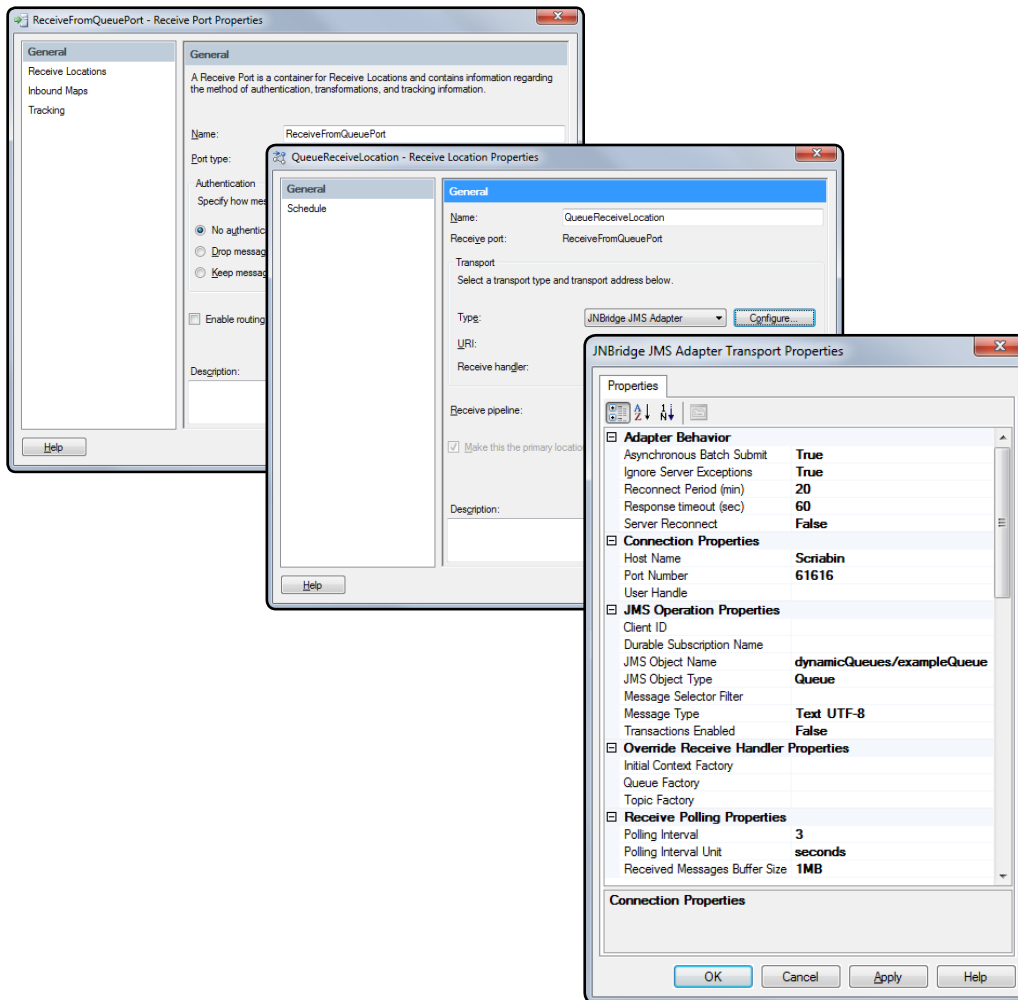


Figure 14. Creating and configuring a Receive Port and Location

Override Transmit Handler Properties Category

Sometimes it may be necessary to override the factory class values entered in the Send Transport Handler properties discussed in the section *JMS Properties*. If the values are not overridden, then the Send Port will use the initial context, queue connection and topic connection factories configured in the send handler properties.

Creating and Configuring Receive Ports and Locations

Creating a Receive Port is similar to creating a Send Port. Simply right-click in an application's Receive Ports pane and choose **New►One-way Receive Port** or **New►Request Response Receive Port**. This will bring up the Receive Port Properties dialog, Figure 14, preceding page.

For more information on Request/Response receive ports, see the section *Messaging Patterns*. Choose Receive Locations in the left-hand pane and select **New** to bring up the Receive Location Properties dialog box. Click on the drop-down list **Type:** and select the JNBridge JMS Adapter. Click the button **Configure** to bring up the Transport Properties dialog, shown in Figure 14, to configure JMS message consumers.

Adapter Behavior Category

■ Asynchronous Batch Submit

If **True**, then each batch will be submitted to the BizTalk Message Box asynchronously. In other words, the submission of each group of messages could overlap if the polling period is short and the batch size is large.

If **False**, each batch is submitted synchronously and the adapter will wait until each batch is successful before submitting the next batch. Set this property to **False** if ordered message reception is required.

! ***The Asynchronous Batch Submit setting is ignored if the receive port is configured for transactions or is a receive/response port. In other words, transactions and receive/response require synchronous behavior.***

■ Ignore Server Exceptions

If this property is **True**, then any asynchronous exceptions sent by the JMS server will be ignored. If **False**, then the receive location will disconnect if *any* asynchronous exception is sent by the JMS server.

■ Server Reconnect

If this property is **False**, then upon receiving an asynchronous exception from the JMS server, the adapter will post an error event, close the connection and auto-disable the receive location. If **True**, then the receive location will post an event, close the connection, but will

remain enabled and periodically attempt a re-connection. If the property Ignore Server Exceptions is True, then this property is ignored.

- **Reconnect Period (minutes)**

If the property Server Reconnect is True, then the receive location will remain enabled and attempt to continually reconnect with a period equal to the entered value. If Server Reconnect is False, this value is ignored. If the property Ignore Server Exceptions is True, this property is also ignored. If the reconnect attempts fail, the failures are posted to the error log, only. If the reconnect attempt succeeds, an event is posted and the location begins to consume messages.

- **Response time-out**

The time-out, in seconds, for a response to arrive for a request. This value is ignored if the receive port is one-way.

Connection Properties Category

Figure 14 shows the category Connection Properties. These properties determine where the JMS server resides and the port number where it is listening for connections.

- **Host Name**

This a text-editable field. Enter the name or IP address of the machine hosting the JMS server.

- **Port Number**

This is a text-editable field. Enter the port number on which the JMS server is accepting connections.

- **User Handle**

This is a text-editable field. The string is used to differentiate the BizTalk URL that represents a receive location. Because the BizTalk send and receive location namespace is flat, the user-defined string allows multiple receive locations to consume messages from the same queue or topic. This provides a mechanism to support concurrent message consumption using message selectors.

JMS Operation Properties Category

These properties determine what operation type of operation the send port will enable.

- **JMS Object Name**

This is a text-editable field. Enter the JNDI name bound to the JMS queue or topic. Depending on the requirements of the JMS implementation, it may be necessary to supply a complete path, e.g. `java:comp/resource/jms/myQueue`.

■ JMS Object Type

This is a drop-down list containing two values: Queue or Topic.

■ Message Type

This is a drop-down list containing four values: Text, Text UTF-8, Text UTF-16, Text ISO-8859-15, Bytes or Map. If a text type is chosen, then the JNBridge JMS Adapter expects to receive a JMS Text Message. If Bytes is chosen, then the JNBridge JMS Adapter expects to receive a JMS Bytes Message. If Map is chosen, the JNBridge JMS Adapter expects to receive a JMS Map Message.

The Text, Text UTF-8, Text UTF-16 and Text ISO-8859-15 values instruct the adapter to encode the incoming XML document into the correct binary representation complete with a Byte Order Marker for publication to the Biztalk Message Box. This is done because a JMS Text message is, by definition, always encoded as UTF-16 BE (big-endian).

The value Text will be interpreted as UTF-8 and encoded as single byte representation and prefaced with the BOM 0xEFBBBF. Text UTF-8 will be encoded as multi-byte, if necessary, and prefaced with the BOM 0xEFBBBF. Text UTF-16 will be encoded as LE (little-endian) double byte representation and prefaced with the BOM 0xFFFE. Text ISO-8859-15 will be encoded to its single-byte representation, but will not be preceded with a BOM. Choosing the wrong encoding could result in a corrupted message.

■ Client ID

This is a string that uniquely identifies the connection maintained by the receive port to the JMS server. This is only used for topic durable subscriptions.

■ Durable Subscription Name

Durable subscriptions are particular to topics only. A durable subscription for a topic allows consumers to register a name with the JMS server such that whenever a receive port is active, all messages in the topic will be received. In this way, a receive port does not have to be continually connected to receive messages from a topic. A receive port that does not use durable subscriptions must be active and connected in order to subscribe to a topic—any messages published by the topic while a nondurable receive port is not active will not be available to that receive port when it becomes active. This is a text-editable field. Enter the durable subscription name.

■ Message Selector Filter

Message selectors are used by receive ports to filter or select messages from topics and queues based on JMS and custom message header properties.

This is a text-editable field. Enter in a selector expression. The expression is derived from a subset of the SQL92 standard.

■ Transactions

Enable or disable local JMS transactions. When transactions are enabled, any resulting error during reception and submission to the BizTalk Message Box will result in the message being unconsumed and still present on the JMS queue or topic. The cause of the problem and the message id is logged to the Application event log.

! *Enabling receive transactions will slow down performance as the effective batch size is 1.*

! *The Transactions property is ignored if the receive port is request/response.*

Override Transmit Handler Properties Category

Sometimes it may be necessary to override the factory class values entered in the Receive Transport Handler properties discussed in the section *JMS Properties*. If the values are not overridden, then the receive port will use the initial context, queue connection and topic connection factories configured in the receive handler properties.

Receive Polling Properties

This category configures the frequency with which BizTalk Server checks for received messages. The JNBridge JMS Adapter receives messages from queues and topics by implementing a background message listener. As such, messages are received when they arrive and are placed in a cache by the adapter. The Polling Properties determine when cached messages are processed and placed in the BizTalk Message Box.

■ Polling Interval

This is a text-editable field. Enter an integer value representing minutes, seconds or milliseconds.

■ Polling Interval Unit

This is a drop-down list containing three choices: minutes, seconds or milliseconds.

■ Received Messages Buffer Size

This is the size of an internal buffer which holds received messages which have not been processed and submitted to the BizTalk Message Box. The JMS adapter consumes messages on a separate thread and, depending on load and throttling parameters, can quickly consume memory holding un-submitted messages. The buffer size is an upper limit of 1, 2, 4, 8 and 16 MB. If the upper limit is reached, the consuming thread will sleep an amount of time equal to the Polling Interval plus 10%.

Messaging Patterns

The JNBridge JMS Adapter for BizTalk Server supports Request/Response and Solicit/Response messaging patterns using JMS as a transport.

Request Response receive ports

The received JMS message which constitutes the request message must have the following JMS header properties set.

- **JMSReplyTo**

This property must contain the JMS destination that the adapter uses to send the response back to the requestor. The response destination can either be a queue or topic. If the JMSReplyTo property is null, the adapter will throw an exception.

- **JMSCorrelationID**

This property is set in the JMS request message header and is copied to the response's JMS header. The developer may choose to override the incoming correlation id by setting the response message's correlation id explicitly in a Pipeline or Orchestration. If the incoming property is empty, and the response message does not have its correlation id explicitly set by an Orchestration or Pipeline, then the response message's JMSCorrelationID will be set to the message id of the request message.

The JMS adapter supports requests from both queues and topics. If the request message is a JMS Text message, the adapter will respond with a text message. If the request message is a JMS Bytes message, the adapter will respond with a JMS Bytes message.

Solicit Response send ports

The JNBridge JMS Adapter for BizTalk Server creates a JMS message used to solicit a response. The adapter automatically creates and manages a temporary queue and places that destination into the solicitation message's JMSReplyTo header property. The adapter then listens for the response message on that queue. If the solicitation message is a JMS Text message, then the adapter expects a JMS Text message in response. If the solicitation message is a JMS Bytes message, then the adapter expects a JMS Bytes message in response.

! *The adapter supports only queues for solicit response messaging patterns.*

Using Message Properties and Map Messages

A JMS message header contains sets of properties. A property is a combination of a data type, name and a value. The JMS specification defines a set of standard, extended and user-defined properties for a JMS header. The JNBridge JMS Adapter for BizTalk Server provides access within an orchestration or pipeline to the standard, extended and user-defined JMS header properties. Additionally, a set of configuration properties are defined in order to support dynamic send ports. Access to the standard and configuration properties is enabled by referencing the schema assembly `[InstallDir]\JNBridge\JMSAdapters\BTS2006\bin\JNBridgeBTS2006JMSProperties.dll` within a Visual Studio 2005/2008 BizTalk project. This assembly is also installed in the Global Assembly Cache at adapter installation. User-defined or the extended JMS properties are supported only if a custom Property Schema is implemented and deployed by the user.

Within an orchestration, properties are accessed from a Message Assignment shape. Property values can be used in expressions. Some properties can be set such that the outgoing JMS message will contain those properties. Configuration properties can be set to dynamically route the message using dynamic send ports.

Properties, XML Schema and .NET namespaces

The JMS adapter has defined three XML schema namespaces along with three corresponding .NET (XLANG) namespaces for use with properties. The three namespaces encapsulate received properties, configuration properties and send properties and are defined, along with reserved property schemes, in the schema assembly `JNBridgeBTS2006JMSProperties.dll`. User defined properties require a custom BizTalk Property Schema and must use the received properties or send properties namespaces.

JMS Receive Properties

This schema uses the XML name space `http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSRecvProperties` and the .NET/XLANG namespace `JMSRecvProperty`. All properties that are contained in a received JMS message header are placed in this namespace when they are written or promoted to the internal BizTalk message context.

Standard and Extended Properties

The received properties namespace defines the following properties that correspond to the supported standard JMS header properties, Table 1, following page. All standard JMS properties are *promoted* to the BTS message context. Promoted properties can be used for routing.

As shown in Table 1, along with the .NET type of the property (germane if accessed in a pipeline) is a syntactic XLANG expression used in a Rule shape within an orchestration. Receive properties can only be accessed, they cannot be assigned.

Name	.NET Type	XLANG Expression Usage
JMSCorrelationID	String	Message_2(JMSRecvProperty.JMSCorrelationID) == "xyz";
JMSDeliveryMode	Int32	Message_2(JMSRecvProperty.JMSDeliveryMode) > 3;
JMSExpiration	String*	Message_2(JMSRecvProperty.JMSExpiration) == "12467...";
JMSMessageID	String	Message_2(JMSRecvProperty.JMSMessageID) == "9C873...";
JMSPriority	Int32	Message_2(JMSRecvProperty.JMSPriority) < 2;
JMSRedelivered	Boolean	Message_2(JMSRecvProperty.JMSRedelivered) == true;
JMSTimestamp	String*	Message_2(JMSRecvProperty.JMSTimestamp) == "12693...";
JMSType	String	Message_2(JMSRecvProperty.JMSType) == "Text";
JMSDestination	String*	Message_2(JMSRecvProperty.JMSDestination) == "holst";

Table 1. Standard JMS properties defined in receive schema namespace.

The properties JMSExpiration and JMSTimestamp are defined in the JMS 1.1 specification as type java.lang.Long. While this would correspond to .NET type System.Long, 64-bit integers are not supported as promoted properties with BizTalk. As all standard JMS header properties are promoted, the string representation of the 64-bit integer is used instead. The property JMSDestination is a string.

! *The JNBridge JMS Adapter for BizTalk does not support the JMS standard header JMSReplyTo as a property that can be accessed using an XLANG expression. The property is used for Request/Response messaging.*

User-defined properties

All user-defined properties found in a JMS message header are *written* to the BTS message context in the receive XML schema namespace. Table 2 shows the supported Java property types and their corresponding BizTalk (.NET) data type.

Java Type	.NET Type	Comment
java.lang.Float	Single	
java.lang.Long	Int64	64-bit properties cannot be promoted, only written
java.lang.Short	Int16	
java.lang.JavaString	String	
java.lang.Byte	Byte	A Java byte is signed. A .NET byte is unsigned.
java.lang.Boolean	Boolean	
java.lang.Integer	Int32	
java.lang.Double	Double	

Table 2. Supported JMS header types and their .NET equivalent

! *The Java type java.lang.Object is not supported as a property. Properties of type Object will be ignored.*

All user-defined properties in a JMS message header that begin with the tag “PROPRM_” will be *promoted* to the BTS message context. All other user-defined properties are written. If a user-defined property is promoted, the user **must** supply a BizTalk Property Schema for that property. This schema must be deployed to the BizTalk application. If the property schema is not found at run-time, the BTS message persistence will fail and the message will be dropped and an error event will be posted. Figure 15 shows a user-defined BizTalk Property Schema that defines a single property.

```
<?xml version="1.0" encoding="utf-16" ?>
<xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns="http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSRecvProperties"
  targetNamespace="http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSRecvProperties"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:appinfo>
      <b:schemaInfo schema_type="property"
        xmlns:b="http://schemas.microsoft.com/BizTalk/2003" />
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="PROPRM_myProp" type="xs:int">
    <xs:annotation>
      <xs:appinfo>
        <b:fieldInfoPropertyGuid="6bda887d-b81a-4660-91dc-4f146e6c1df6"
          propSchFieldBase="MessageContextPropertyBase" />
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
</xs:schema>
```

Figure 15. A user defined property schema generated in Visual Studio

Notice that the target namespace is the defined receive namespace, `http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSRecvProperties`. All properties found in an incoming JMS header are written or promoted to the BTS message context in this namespace. As such, a user-defined property that has been promoted can be accessed in a Rule form using the XLANG expression:

```
Message_2(MyRecvProperties.PROPRM_myProp) > 4
```

Additionally, promoted properties can be used for routing. Extended JMS header properties, those that begin with “JMSX”, are always written to the BTS message context, they are never promoted.

JMS Send Properties

This schema uses the XML name space `http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSSendProperties` and the .NET/XLANG namespace `JMSSendProperty`. All properties in this namespace, regardless whether written or promoted, are placed in the JMS message header during a transmission from a send port. This namespace schema predefines three standard JMS header properties, Table 3 on the following page, in the assembly `JNBridgeBTS2006JMSProperties.dll`. All properties in the send namespace can be set in a Message Assignment shape within an orchestration.

User properties that will be transmitted in the outgoing JMS message header must be defined in a user supplied BizTalk Property Schema, Figure 16.

Name	.NET Type	Comment
JMSCorrelationID	String	
JMSType	String	
JMSPriority	Int32	May be overwritten by JMS provider

Table 3. Send Properties

Note that the same property with the auto-promote tag, PROPRM_myProp, that was defined in the receive namespace for incoming JMS messages on page 25, is defined again here. As long as the .NET/XLANG namespace is different for each separate xsd file (one for the receive XML namespace and one for the send XML namespace), XLANG expressions that “get” receive properties are separate from expressions that “set” the same properties. Even if the send namespace contains several properties, only those set in a Message Assignment shape will be written to the outward bound JMS message. Figure 17, following page, displays a BizTalk Expression Editor setting the outgoing JMSCorrelationID as well as incrementing the user-defined property, PROPRM_myProp. Because received messages are immutable, the initial expression copies Message_1, the received inbound message, to Message_2, the outbound message.

```

<?xml version="1.0" encoding="utf-16" ?>
- <xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns="http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSSendProperties"
  targetNamespace="http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSSendProperties"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <xs:annotation>
- <xs:appinfo>
  <b:schemaInfo schema_type="property" xmlns:b="http://schemas.microsoft.com/BizTalk/2003" />
</xs:appinfo>
</xs:annotation>
- <xs:element name="JMSXGroupID" type="xs:string">
- <xs:annotation>
- <xs:appinfo>
  <b:fieldInfo propertyGuid="891ad509-b8ed-40c8-92f4-3587b3a55ba4"
    propSchFieldBase="MessageContextPropertyBase" />
</xs:appinfo>
</xs:annotation>
</xs:element>
- <xs:element name="PROPRM_myProp" type="xs:int">
- <xs:annotation>
- <xs:appinfo>
  <b:fieldInfo propertyGuid="fa07e569-1d92-42b7-8fdf-e3f32260a283"
    propSchFieldBase="MessageContextPropertyBase" />
</xs:appinfo>
</xs:annotation>
</xs:element>
</xs:schema>

```

Figure 16. Send BizTalk Property Schema

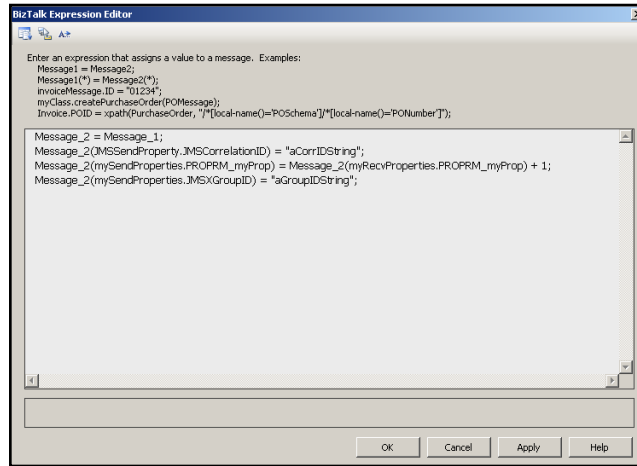


Figure 17. XLANG Expressions setting outbound properties

JMS Configure Properties

This schema uses the XML name space `http://schemas.jnbridge.com/JMSAdapter/BizTalk/JMSConfProperties` and the .NET/XLANG namespace `JMSConfProperty`. This namespace schema predefines five properties that can be set by the user in a Message Assignment shape. If the orchestration uses a dynamic send port, the configure properties can be used to dynamically route messages. Table 4, on the next page, lists the configure properties. The properties are defined in the assembly, `JNBridgeBTS2006JMSProperties.dll`.

Name	.NET Type	Comment
DS_Hostname	String	
DS_Port	Int32	
DS_DestType	String	Must be "Topic" or "Queue". Case-sensitive.
DS_Destination	String	The JNDI location of the topic or queue.
DS_JMSType	String	Must be "Text" or "Binary". Case-sensitive.

Table 4. Dynamic send port configuration properties.

Using Dynamic Send Ports

The JMS configuration namespace can be used with dynamic send ports. Dynamic send ports are configured as dynamic within an orchestration, hence when the orchestration is deployed to BizTalk Server, there is no bound logical port. The binding of a transport and an address to the dynamic port is done within the orchestration. Figure 18, below, shows a portion of an orchestration configuring a dynamic send port.

The expressions in the Message Assignment form sets the five configure properties. The Expression form binds the JNBridge JMS transport to the dynamic send port by using the expression:

```
Port_2(Microsoft.XLANGs.BaseTypes.Address) = "jms://";
```

By using the registered transport URL, `jms://`, for the JNBridge JMS Adapter, BizTalk Server knows to bind the adapter to the dynamic send port.

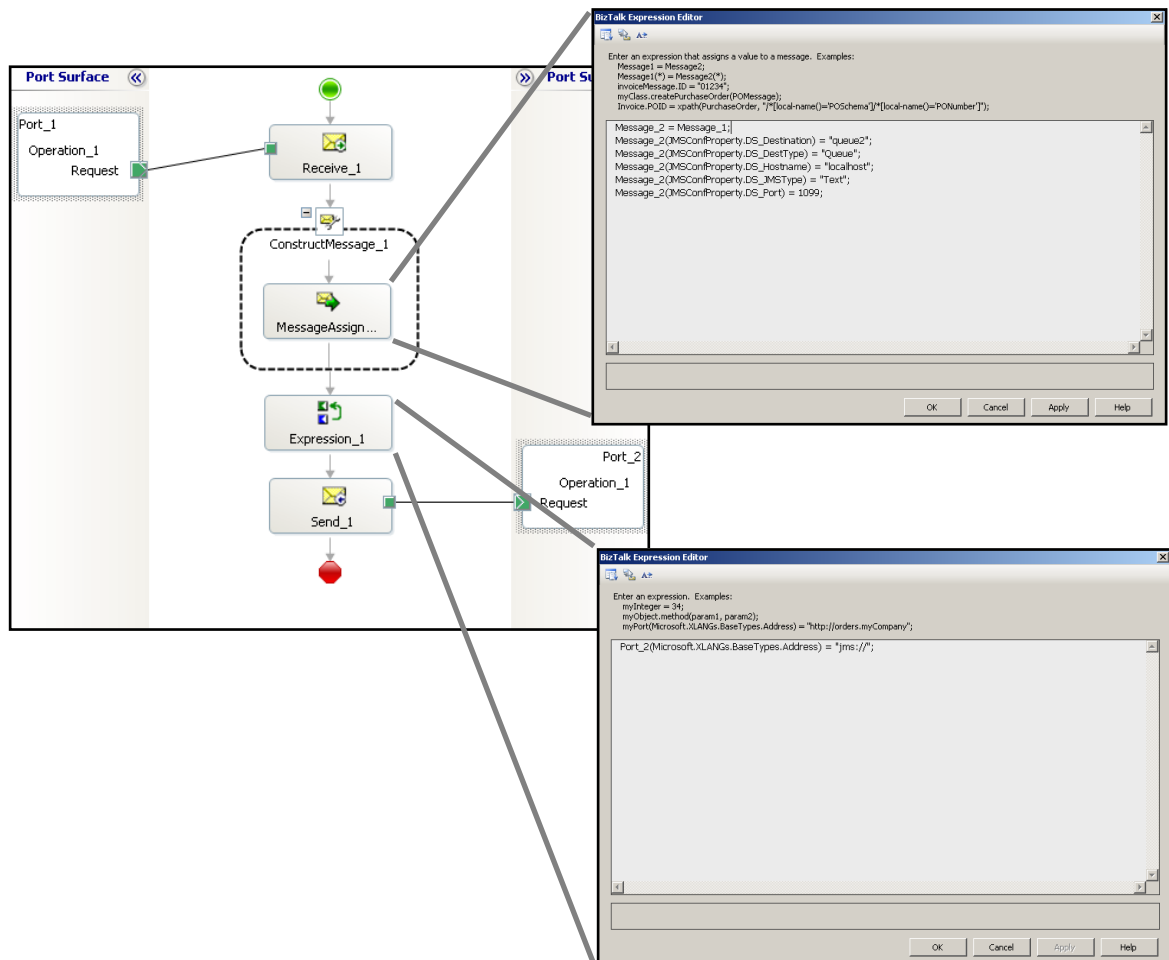


Figure 18. Configuring a dynamic send port in an orchestration

USERS' GUIDE

Using Map Messages

The adapter will receive JMS Map Messages. A JMS Map Message is a message whose body contains a set of custom properties. When a JMS Map Message is received, the adapter will convert the map into a very simple XML document, see Figure 19.

```
<?xml version="1.0" encoding="utf-16"?>
<Map>
  <shortValue type="Short">32767</shortValue>
  <minFloatValue type="Float">-3.402823E+38</minFloatValue>
  <longValue type="Long">9223372036854775807</longValue>
  <maxDoubleValue type="Double">1.79769313486232E+308</maxDoubleValue>
  <stringValue type="String">The quick brown fox...</stringValue>
  <byteValue type="Byte">36</byteValue>
  <byteArrayValue type="Bytes">VABoAGUAIABxAHUA=</byteArrayValue>
  <minDoubleValue type="Double">-1.79769313486232E+308</minDoubleValue>
  <maxFloatValue type="Float">3.402823E+38</maxFloatValue>
  <booleanValue type="Boolean">true</booleanValue>
  <charValue type="Char">12</charValue>
  <intValue type="Int">2147483647</intValue>
</Map>
```

Figure 19. A Map Message converted to XML

It will be up to the user to provide the necessary schema for the particular properties contained in a Map Message.

Trouble Shooting

Deploy JMS Header Schema to BizTalk Application

Where did my messages go? Receiving JMS messages from a JMS destination will cause this error if the JMS properties schema has not been deployed to BizTalk. Even if you reference the schema assembly in a BizTalk Visual Studio project, the schema assembly will not be deployed.

“Missing or malformed schema for a promoted property. Message cannot be re-submitted: unable to persist message in BTS DB. Did you deploy [%INSTALL_DIR%]JNBridgeJMSAdapters\BTS2006\bin\JNBridgeBTS2006JMSProperties.dll to your BTS application?”

In order to properly handle JMS header properties within BizTalk, you must deploy the assembly, `JNBridgeBTS2006JMSProperties.dll`, to your BizTalk application. This assembly contains the XSD namespaces and schemas used by the JNBridge JMS Adapter to promote JMS header properties within messages stored in the BizTalk Message Box.

! *Deploying this assembly is mandatory.*

■ To deploy the schema assembly

- 1 Open up the BizTalk Administrator and open your BizTalk application in the left-side tree view.
- 2 Right click on the application's root node and choose **Add►Resources**. This opens the **Add Resources** dialog.
- 3 In the dialog, click on the **Add** button and navigate to the schema DLL in the adapter install directory, e.g. `C:\Program Files\JNBridge\JMSAdapters\BTS2006\bin\JNBridgeBTS2006JMSProperties.dll`.
- 4 Click on **OK** to close the **Add Resources** dialog.
- 5 Open the **Schemas** folder in your application. You should see the three schemas:
`JMSSendProperty.SendPropertySchema`, `JMSRecvProperty.RecvPropertySchema`
and `JMSConfProperty.ConfPropertySchema`.
- 6 Restart the host instance and application.