



**Using the JNBridge JMS Adapter for BizTalk Server
with JBoss AS
version 4.0**

www.jnbridge.com

Using the JMS Adapter with JBoss

JNBridge, LLC
www.jnbridge.com

COPYRIGHT © 2008-2016 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 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.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Contents

Quick Config for JBoss AS	4
Adapter Transport Handler Properties.....	4
Adapter Send or Receive Port Properties	5
Using the JMS Adapter with JBoss AS	6
Resources.....	6
BizTalk Machine Prerequisites	6
Configuring the Adapter Send and Receive Transport Handlers.....	7
JMS Properties Category	7
JNBridge Properties Category.....	9
Security Properties Category	11
Configuring Send Ports and Receive Locations	12
Configuring Send Ports.....	12
Connection Properties Category	12
JMS Operation Properties Category	12
Configuring Receive Ports and Locations.....	13
Connection Properties Category	13
JMS Operation Properties Category	14
JNDI Names	15
Trouble Shooting.....	16
Unable to connect to the JBoss server.....	16
Unable to cast object	16

Using the JMS Adapter with JBoss

Quick Config for JBoss AS

Version numbers refer to open source community releases. The latest version number concurrent with publication is used. It is presumed that the configuration for the current version will remain germane for succeeding versions, at least for a while.

Adapter Transport Handler Properties

- Initial Context Factory

For JBoss AS 7.1.1: `org.jboss.naming.remote.client.InitialContextFactory`

For all earlier versions: `org.jnp.interfaces.NamingContextFactory`

- JMS Scheme

For JBoss AS 8 and greater: `http-remoting`

For JBoss AS 7.1.1: `remote`

For all earlier versions: `jnp`

- Queue Connection Factory (default)

For JBoss AS 7.1.1: `jms/RemoteConnectionFactory`

For all earlier versions: `ConnectionFactory`

- Topic Connection Factory

For JBoss AS 7.1.1 and greater: `jms/RemoteConnectionFactory`

For all earlier versions: `ConnectionFactory`

- Class Path

For JBoss AS 7.1.1 and greater: `jboss-client.jar`

For JBoss version 6.x, HornetQ is the messaging provider, so use these JAR files:

```
... \common\lib\hornetq-bootstrap.jar
... \common\lib\hornetq-core.jar
... \common\lib\hornetq-jboss-as-integration.jar
... \common\lib\hornetq-jms.jar
... \common\lib\hornetq-logging.jar
```

For JBoss versions 5.x, use this jar file. Note that this jar file contains an embedded class path, so it must remain *in situ* with all other JBoss jar files: `jbossall-client.jar`

For JBoss versions 4.3.x, use these jar files:

```
jboss-messaging-client.jar, jbossall-client.jar, javassist.
```

`jar, trove.jar, log4j.jar, jboss-aop-jdk50.jar`

For JBoss versions 4.2.x and greater, please use these jar files.

`jbossall-client.jar, javassist.jar, trove.jar, log4j.jar, jboss-aop-jdk50.jar`

For JBoss versions 4.0.x, use this jar file:

`jbossall-client.jar`

- JVM Path (example)

`C:\Program Files\Java\jre7\bin\client\jvm.dll`

Adapter Send or Receive Port Properties

- Port Number

For JBoss AS 7.1.1: `4447`

For all earlier versions: `1099`

- JMS Object Name

For JBoss AS 7.1.1: `jms/queue/test`

For all earlier versions: `queue/testQueue`

Using the JMS Adapter with JBoss

Using the JMS Adapter with JBoss AS

This document uses the example JMS service that comes pre-configured in JBoss. This document assumes some passing familiarity with enterprise Java such as JNDI contexts, factories and general connection requirements and parameters. This document also assumes that the BizTalk developer has access to particular information peculiar to the target JMS implementation. Finally, this document assumes that the reader is knowledgeable and experienced with BizTalk Server .

The configuration in this document is intended for both the open source version of JBoss as well as the subscription version from Red Hat. The version numbers refer to the JBoss community releases, not the Red Hat versions. There will be differences in the JAR files—the class paths—between the Red Hat and Community versions, as well as differences between seemingly minor releases. The latest version number concurrent with publication is used here, 7.1.1. It is presumed that the configuration for the current version will remain germane for succeeding versions, at least for a while.

Please check documentation at either Red Hat or at www.jboss.org.

Resources

- The user guide, *Using the JNBridge JMS Adapter for BizTalk Server*.
- Chances are, if the target JMS implementation is mature, the values for the configuration of BizTalk transport handlers and send/receive ports can be supplied by the JBoss administrator, developers or gleaned from existing JMS client code.
- If the JBoss AS implementation targeted is not configured, then the default example JMS service installed with JBoss can be used for proof-of-concept evaluations.

BizTalk Machine Prerequisites

The following prerequisites are needed for the adapter.

- A public Java Run-time Environment (JRE) must be installed on the target machine. The JNBridge JMS Adapter supports the Standard Edition JRE 7 or above.

- The JNBridge JMS Adapter for .NET uses a stand-alone client JMS environment supplied by JBoss. This stand-alone client consists of one to six jar files depending on the JBoss version. Please see the section, *Class Path*, below

Deploy JMS Header Schema to BizTalk 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.

Configuring the Adapter Send and Receive Transport Handlers

The transport handler property grids for the Send and Receive sides contain properties global to all send or receive ports configured to use the JNBridge JMS Adapter and that reside in the same BTS host instance. In other words, all JMS Adapter send or receive ports in the BTS host instance 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.

JMS Properties Category

The JMS Properties category are properties used to properly connect to a JMS server.

Using the JMS Adapter with JBoss

■ 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 a default JBoss connection factory, `AUTO_ACKNOWLEDGE` is the default configuration.

■ Initial Context Factory

This is a text-editable field containing the name of the JNDI initial context factory. The initial context factory is a JNDI class used to locate and instance factories and JMS destinations. The default initial context factory for JBoss AS 7.1.1 is:

```
org.jboss.naming.remote.client.InitialContextFactory
```

For all earlier versions: `org.jnp.interfaces.NamingContextFactory`

! *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 RMI implementation. The protocol is part of the URI used to connect to the JMS service.

For JBoss AS 8 and greater: `http-remoting`

For JBoss 7.1.1, the scheme is: `remote`

For all earlier versions: `jnp`

■ JMS Version

This property tells the adapter which JMS implementation to expect when it loads the vendor's client stack—the JAR files in the Class Path property. The drop-down list contains two choices, 'JMS 1.1' and 'JMS 2.0'.

■ 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, simple and strong. If the choice is simple, then the server expects a user name and password. If the choice is strong, then the server expects a digital certificate security strategy, usually based on the X.509 standard.

! *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.*

■ Queue Connection Factory

This is a text-editable field. The default queue connection factory in JBoss AS 7.1.1 and greater is:

`jms/RemoteConnectionFactory`

For all earlier versions: `ConnectionFactory`

! *This value includes the JNDI directory path to the connection factory. The complete JNDI path must be provided.*

■ Topic Connection Factory

This is a text-editable field. The default queue connection factory in JBoss AS 7.1.1 and greater:

`jms/RemoteConnectionFactory`

For all earlier versions: `ConnectionFactory`

! *This value includes the JNDI directory path to the connection factory. The complete JNDI path must be provided.*

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. Note that only checked elements will be added to the class path when the dialog is dismissed.

For JBoss AS version 7.1.1:

`[JBOSS_HOME]\bin\client\jboss-client.jar`

For JBoss version 6.x, HornetQ is the messaging provider:

`[JBOSS_HOME]\common\lib\hornetq-bootstrap.jar`

`[JBOSS_HOME]\common\lib\hornetq-core.jar`

Using the JMS Adapter with JBoss

```
[JBoss_HOME]\common\lib\hornetq-jboss-as-integration.jar
[JBoss_HOME]\common\lib\hornetq--jms.jar
[JBoss_HOME]\common\lib\hornetq-logging.jar
```

JBoss version 5.x

```
[JBoss_HOME]\client\jbossall-client.jar
```

This JAR file contains an embedded classpath in its manifest that points to many of the JAR files in the `client` directory. As such, it is recommended that the entire `client` directory be copied to the development or deployment machine.

JBoss versions 4.3.x and above

```
[JBoss_HOME]\client\jboss-messaging-client.jar
[JBoss_HOME]\client\jbossall-client.jar
[JBoss_HOME]\client\log4j.jar
[JBoss_HOME]\server\deploy\deploy\jboss-aop-jdk50.deployer\jboss-
aop-jdk50.jar
[JBoss_HOME]\client\javassist.jar
[JBoss_HOME]\client\trove.jar
```

These six JAR files can be copied from the above location in the JBoss directories to a convenient location on the development or deployment machine.

JBoss versions 4.2.x and above

```
[JBoss_HOME]\client\jbossall-client.jar
[JBoss_HOME]\client\log4j.jar
[JBoss_HOME]\server\<SERVER_NAME>\deploy\jboss-aop.deployer\
jboss-aop.jar
[JBoss_HOME]\server\<SERVER_NAME>\lib\javassist.jar
[JBoss_HOME]\server\<SERVER_NAME>\lib\trove.jar
```

These five JAR files can be copied from the above location in the JBoss directories to a convenient location on the development or deployment machine.

JBoss versions 4.0.x

```
jbossall-client.jar
```

■ 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. In the example shown, the JRE used is:

```
C:\Program Files\Java\jre7\bin\client\jvm.dll
```

Security Properties Category

This category provides security credentials necessary to connect to a JMS server, if the JMS implementation supports security mode simple.

- **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.

Configuring Send Ports and Receive Locations

Configuring Send Ports

Connection Properties Category

These properties determine where the JMS server resides and the port number where it is listening for connections.

- Host Name

This is 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. For a JEE server this is usually a IIOP, CORBA or RMI bootstrap port. For JBoss, this port is usually, by default, **1099**, for the jnp protocol.

For JBoss versions (7.1, or higher), the port is, by default, **4447**

- Proprietary Connection String

This is a text-editable field. This property is only used if the JMS implementation uses complex URLs containing query expressions, or some proprietary connection string, that can not be constructed from the Host Name and Port Number properties. If this property contains a value, then the Host Name and Port Number properties will be ignored.

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. For JBoss JMS, the example queue or topic is at:

`queue/testQueue`

`topic/testTopic`

`jms/queue/test` (JBoss AS 7.1)

- **JMS Object Type**

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

- **Message Type**

This is a drop-down list containing the values: Text, Text UTF, Text ISO-8859-15 or Bytes.

If Text is chosen, then the JNBridge JMS Adapter will send a JMS Text Message. Because a JMS Text Message is by definition UTF-16 BE, the types Text UTF and Text ISO-8859-15 ensure that the string which is the payload of the JMS message is correctly encoded from the binary representation in the BizTalk Message Box. If the type Text is chosen, then the binary representation is considered UTF-8. If Bytes is chosen, then the JNBridge JMS Adapter will send a JMS Bytes Message.

Configuring Receive Ports and Locations

Connection Properties Category

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. For a JEE server this is usually a IIOP, CORBA or RMI bootstrap port. For JBoss, this port is usually, by default, **1099**.

- **Proprietary Connection String**

This is a text-editable field. This property is only used if the JMS implementation uses complex URLs containing query expressions, or some proprietary connection string, that can not be constructed from the Host Name and Port Number properties. If this property contains a value, then the Host Name and Port Number properties will be ignored.

Using the JMS Adapter with JBoss

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. For JBoss JMS, the example topic or queue is at:

`queue/testQueue`

`topic/testTopic`

■ JMS Object Type

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

■ Message Type

This is a drop-down list containing the values: Text, Text UTF-8, Text UTF-16, Text ISO-8859-15, Bytes or Map. If Text 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, then the JNBridge JMS Adapter expects to receive a JMS Map Message. Because a JMS Text Message by definition contains UTF-16 BE text, the types Text UTF-8, Text UTF-16 and Text ISO-8859-15 ensure that the text in the body of the message is encoded correctly to binary for submittal to the BizTalk Message Box. The type Text without a qualifier means UTF-8.

■ Client ID

This is a unique string that identifies the receive port connection to JBoss. It is only used if durable subscriptions are enabled.

■ 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.

Using the JMS Adapter with JBoss

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

JNDI Names

Figure 1 shows the JBoss Global JNDI Namespace. This listing can be obtained by invoking the `list()`

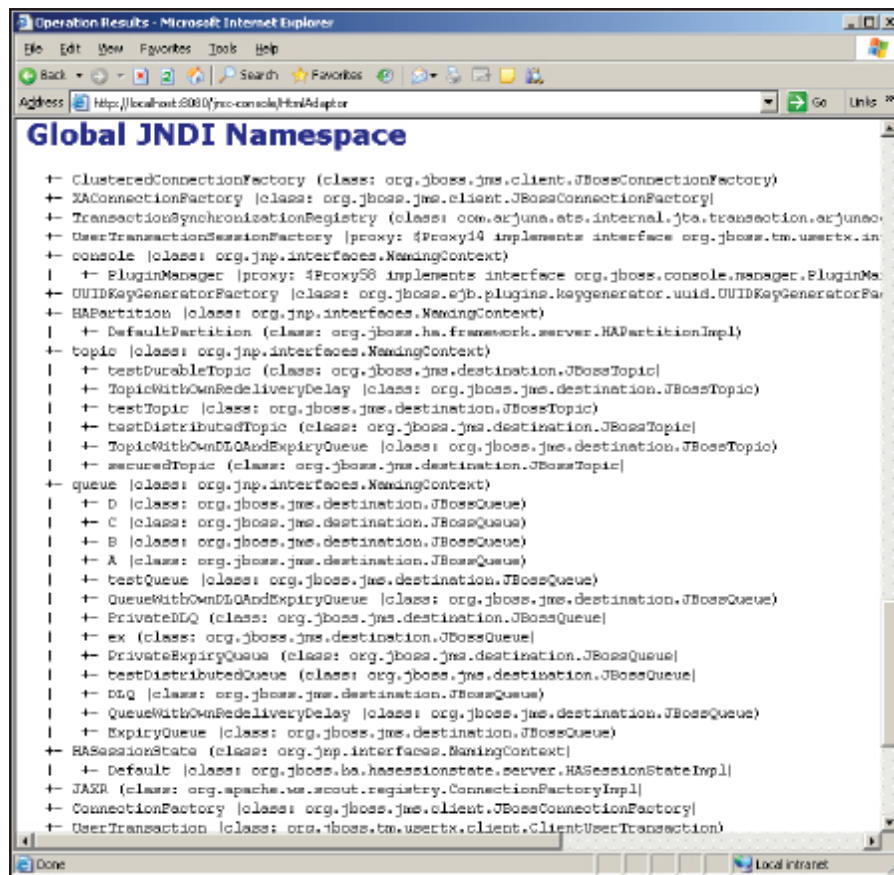


Figure 1. The JNDI tree

Using the JMS Adapter with JBoss

method of the JMX MBean JNDIView service from the JBoss jmx-console. The list is displaying the JNDI directory structure for the initial JBoss installation. It is important to use complete JNDI paths, either forward slash ('/') or dot('.') delineated, for connection factories and JMS destinations.

Trouble Shooting

This section discusses issues and exceptions and their work arounds.

Unable to connect to the JBoss server

This problem can manifest as several exceptions all which indicate that a connection could not be made. Make sure that the machine name and port are correct. Use `telnet.exe` to try and connect to the machine and port as a test.

JBoss connection security is based on allowable client IP addresses. JBoss out-of-the-box only allows connections from the loopback interface, i.e. localhost (127.0.0.1). Starting JBoss with the argument `-b0.0.0.0` will allow connections from any IP address.

Unable to cast object

Depending on the version of JBoss, the following exception, or one like it, may be thrown at start-up or when first accessing a queue or topic:

```
Unable to cast object of type 'javax.naming.Reference' to type 'javax.jms.Queue'. Exception : Method 'getReference' in type 'org.jboss.mq.SpyQueue'
```

This is caused by the JBoss class loader isolation mechanism in JNDI naming environments. It can be solved by appending the JAR files to the JVM boot classpath. This can be done by using the **JVM Arguments** property in the transport handler property sheet. Use this syntax.

```
-Xbootclasspath/a:[Class Path]
```

Where [Class Path] is the same class path as the value for the **Class Path** property.

It is also recommended that each send port or receive location is isolated in its own host instance.

