



**Using the JNBridge JMS Adapter for .NET
with IBM WebSphere
version 4.0**

www.jnbridge.com

Using the JMS Adapter with WebSphere

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 WebSphere	4
Binding Properties Tab	4
URI Properties Tab	5
Using the JMS Adapter with WebSphere	6
Machine Prerequisites	6
Installing the IBM WebSphere JMS client	7
Obtaining the IBM WebSphere 7/8 JMS client	7
Configuring the Sun ORB	7
Configuring the Adapter Connection Properties	7
Binding Properties Tab	8
JNBridge Properties Category	9
URI Properties Tab	11
Security Tab	11
JNDI Names	12
Using the RefFSContextFactory JNDI with WebSphere MQ	13

Using the JMS Adapter with WebSphere

Quick Config for WebSphere

The proper configuration of the adapter for WebSphere JMS is dependent on the presence of the WebSphere Application Server (WAS). The application server provides JMS as a generic service, abstracting the underlying messaging provider. This means that regardless of the underlying messaging provider, WAS provides a consistent configuration for the JMS service. This is true even if the underlying messaging provider in WAS is WebSphere MQ (WMQ).

If WAS is not being used, then WMQ is being used as a *stand-alone* broker—there is no application server. In that case, the configuration will be different. Notably, the WMQ configuration will be dependent on the Java Naming and Directory Interface (JNDI) used by WMQ. JNDI is required by the adapter. A JNDI-like scaffolding is used by WMQ. The JMS adapter supports the file-based JNDI based on the Sun *RefFSContext* implementation. This support requires the use of a wrapper that extends the *RefFSContextFactory* class. For more information, see the section *Using the RefFSContextFactory JNDI with WebSphere MQ* at the end of this document.

Binding Properties Tab

■ Initial Context Factory

Using WAS: `com.ibm.websphere.naming.WsnInitialContextFactory`

Using stand-alone WMQ: `com.sun.jndi.fscontext.JnbBtsRefFSContextFactory`

■ JMS Scheme

Using WAS: `iiop`

Using stand-alone WMQ: `fscontext`

■ Queue Factory

No connection factories are pre-configured in WAS or WMQ. Please configure connection factories and use the defined JNDI path as the value for this property.

■ Topic Factory

No connection factories are pre-configured in WAS or WMQ. Please configure connection factories and use the defined JNDI path as the value for this property.

■ Class Path

For WebSphere 6

IBM JRE: `sibc.jndi.jar, sibc.jms.jar`

Oracle/Sun JRE: `sibc.jndi.jar, sibc.jms.jar, sibc.orb.jar`

Using the JMS Adapter with WebSphere

For WebSphere 7 and 8 (for version 8, use JAR files with ‘_8.x.x’):

IBM JRE:

```
com.ibm.ws.sib.thin.jms_7.0.0.jar  
com.ibm.ws.ejb.thinclient_7.0.0.jar
```

Oracle/Sun JRE:

```
com.ibm.ws.sib.thin.jms_7.0.0.jar  
com.ibm.ws.ejb.thinclient_7.0.0.jar  
com.ibm.ws.orb_7.0.0.jar
```

Using WMQ:

```
com.ibm.mqjms.jar  
com.ibm.mq.jar  
jnbjms-fscontext.jar
```

■ **JVM Path (example)**

Using IBM JRE: `C:\Program Files\IBM\WebSphere\AppServer\java\jre\bin\j9vm\jvm.dll`

Using Oracle/Sun JRE: `C:\Program Files\Java\jre7\bin\client\jvm.dll`

■ **JVM Argument**

For WMQ (only):

```
-Djava.naming.provider.url=file:/C:/[PathToRefFSContextBindingFile]
```

URI Properties Tab

■ **Port Number**

Using WAS: **2809**

Using WMQ: **1414**

Using the JMS Adapter with WebSphere

Using the JMS Adapter with WebSphere

This document demonstrates connecting the JNBridge JMS Adapter for .NET to the IBM WebSphere Application Server. The configuration discussed here is applicable to WebSphere releases 6/7/8. Within WebSphere, the underlying JMS service can be provided by the default messaging provider bundled with WebSphere or a WebSphere MQ implementation. Within WAS, the messaging provider that supplies the JMS service will always require the WAS connection configuration regardless of the underlying messaging provider, including WMQ. The configuration for stand-alone WMQ means that WAS is not involved.

WebSphere does not come with a pre-configured JMS provider. It is necessary to configure the service using the default messaging provider or WMQ. 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 .NET developer has access to particular information peculiar to the target JMS implementation. Finally, this document assumes that the reader is knowledgeable and experienced with .NET.

This document only discusses those property values that pertain to communicating with WebLogic. Other properties that are not discussed here can be found in the companion *Using the JNBridge JMS Adapter for .NET* document.

Resources

- The user guide, *JNBridge JMS Adapter for .NET Users' Guide*.
- This link is IBM documentation providing an overview and download of the stand-alone [WebSphere 6 JMS client](#) required by the JMS adapter. This link provides information about [installing and configuring](#) the stand-alone client. This link provides information on the [WebSphere 7/8 JMS thin client](#).
- It is strongly suggested that the developer read the section *Tips and Tricks* in the *JNBridge JMS Adapter for .NET Users' Guide*.

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 stand-alone WebSphere client can use either the IBM Java Run-time Environment (JRE) or the Oracle/Sun JRE. The IBM JRE ships with WebSphere. It can also be obtained by downloading the [IBM Eclipse Development](#) environment. Please contact IBM for additional licensing information regarding the IBM JRE. The Oracle/Sun JRE can be downloaded from the Sun website. Regardless of which JRE is used, it must be present on the BizTalk machine. Supported JRE versions are Java 7.

Using the JMS Adapter with WebSphere

Configuring the .NET adapter machine

Installing the IBM WebSphere JMS client

The JNBridge JMS Adapter for .NET uses the WebSphere stand-alone JMS environment supplied by IBM (see *Machine Prerequisites*). This JMS client can be downloaded as a single JAR file, `sibc_install-<version>.jar`, from the IBM website.

If the IBM JRE is being used, you must install the environment by executing this command:

```
java -jar sibc_install-<version>.jar jms_jndi_ibm [InstallDir]
```

If the Sun JRE is being used, use this command line:

```
java -jar sibc_install-<version>.jar jms_jndi_sun [InstallDir]
```

The installation supporting the IBM JRE will result in two JAR files, `sibc.jndi.jar` and `sibc.jms.jar`, being installed in `[InstallDir]\lib`. The installation supporting the Sun JRE will add a third JAR file, `sibc.orb.jar`.

Obtaining the IBM WebSphere 7/8 JMS client

Starting with WebSphere 7, IBM no longer supplies a download of the stand-alone JMS client. The client JAR files must be obtained from the WebSphere 7 or 8 installation, e.g., `C:\Program Files\IBM\WebSphere\AppServer\runtimes` (see *BizTalk Server Machine Prerequisites*).

Configuring the Sun ORB

If the Sun JRE is being used, then a required JEE system property configuring the Sun Object Request Broker (ORB) must be supplied to the Sun JVM. If the IBM JRE is being used, skip this configuration step. Arguments are passed to the Sun JVM by the adapter using the JVM Arguments property. Put this argument in the property `JVM Arguments` in the Binding Properties tab or as the value for the element `JVMArgs` in the `app.config` file.

```
-Dcom.ibm.CORBA.ORBInit=com.ibm.ws.sib.client.ORB
```

Configuring the Adapter Connection Properties

The **Add Adapter Service Reference** development tool in Visual Studio is used to generate the `app.config` file and the WCF client file. The WCF client contains the methods chosen to send and receive JMS messages. The `app.config` file contains the binding element whose attribute values are the parameters used to initialize and connect to the JMS server. Each of the binding attribute values can be entered in the **Binding Properties** tab of the **Configure Adapter** dialog box. While it is possible to enter these values and then connect to the JMS server from the **Add Adapter Service Reference** dialog box, it is more efficient to work off-line and generate the `app.config` and the WCF client using only the generic operations. Please see the section *Tips and Tricks* in the *JNBridge JMS Adapter for .NET Users' Guide* for more information about working off-line and using the generic operations.

Using the JMS Adapter with WebSphere

What follows are the property values required to connect to the JMS implementation in WebSphere. The versions of WebSphere discussed in this document are v6 and v7.

Binding Properties Tab

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

■ Choose JMS Vendor

This is a drop-down control that comes pre-charged with default vendor connection properties. Click and select *WebSphere* and the default configuration values will automatically appear.

■ Custom 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 and Port properties in the URI tab will be ignored.

■ 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 WebSphere connection factory, `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 class used to create a JNDI initial context used to look-up connection factories and destinations. The default initial context factory for WebSphere Application Server is:

```
com.ibm.websphere.naming.WsnInitialContextFactory
```

If WebSphere MQ is being used stand-alone, the default initial context factory is:

```
com.sun.jndi.fscontext.JnbBtsRefFSContextFactory
```

! *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 WebSphere Application Server, the scheme is:

Using the JMS Adapter with WebSphere

`iiop.`

If WebSphere MQ is being used stand-alone, the scheme is:

`fscontext`

■ 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 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. No connection factories are pre-configured in WebSphere.

! *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. No connection factories are pre-configured in WebSphere.

! *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 to .NET interoperability

Using the JMS Adapter with WebSphere

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 WebSphere Application Server 6

IBM JRE: `sibc.jndi.jar`, `sibc.jms.jar`

Oracle/Sun JRE: `sibc.jndi.jar`, `sibc.jms.jar`, `sibc.orb.jar`

For WebSphere Application Server 7 and 8 (for version 8, use JAR files with ‘_8.x.x’):

IBM JRE:

`com.ibm.ws.sib.thin.jms_7.0.0.jar`

`com.ibm.ws.ejb.thinclient_7.0.0.jar`

Oracle/Sun JRE:

`com.ibm.ws.sib.thin.jms_7.0.0.jar`

`com.ibm.ws.ejb.thinclient_7.0.0.jar`

`com.ibm.ws.orb_7.0.0.jar`

For WebSphere MQ stand-alone 7 and 8:

`com.ibm.mqjms.jar`

`com.ibm.mq.jar`

`jnbjms-fscontext.jar`

Note that the WMQ JAR file, `com.ibm.mqjms.jar`, contains an embedded classpath that points to several other JAR files in the complete WMQ installation. This classpath changes with each release of WMQ. A JAR file is just a zip archive of a directory structure, so the manifest (META-INF/MANIFEST.MF) can be opened with a good zip utility in order to look at the classpath, or change qualifier ‘.jar’ to ‘.zip’ and use Windows Explorer. By doing this, you can scrape the necessary JAR files from the WMQ install (`[WMQ InstallDir]\java\lib`) and place them, along with the three above, on the client machine.

The JAR file shipped with the adapter, `jnbjms-fscontext.jar`, can be found in the **support** directory in the zip archive `RefFSContextFC.zip`.

■ 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:

Using the JMS Adapter with WebSphere

Using IBM JRE: `C:\Program Files\IBM\WebSphere\AppServer\java\jre\bin\j9vm\jvm.dll`

Using Oracle/Sun JRE: `C:\Program Files\Java\jre7\bin\client\jvm.dll`

■ JVM Argument

For WMQ (only), please note the use of forward strokes:

```
-Djava.naming.provider.url=file:/C:/[PathToRefFSContextBindingFile]
```

URI Properties Tab

This category provides the location of the host where WebSphere is running and the port that WebSphere listens to for connections.

■ Host

The host name or IP address of the machine running WebSphere.

■ Port

This is a text editable field. Enter the port where WebSphere is listening for client connections. By default this is port **2809**.

Security Tab

This tab need only be used if security is implemented in the WebSphere server and is of type *simple*. Note that the password will appear in clear text in the app.config file. Please see the section *Deploying Solutions* in the *Users' Guide*.

■ Client credential type

This is a drop-down control. Choose the credential type *Username*.

■ User name

Enter the user name credential.

■ Password

Enter the password credential

Using the JMS Adapter with WebSphere

JNDI Names

Figure 1 shows the WebSphere 6.1 Integrated Solutions Console. The console is displaying the configured JMS queues. Note that there are two configured queues, one each for two different JMS providers. The queue, `test_Q`, is created under the default messaging provider (the JMS implementation that comes with WebSphere) and has the JNDI name, `myQ`. The queue, `MQ_Q`, is provided by MQSeries and has the JNDI name, `myMQ_Q`.

It is important to use complete JNDI paths, either forward slash (‘/’) or dot (‘.’) delineated, for connection factories and JMS destinations.

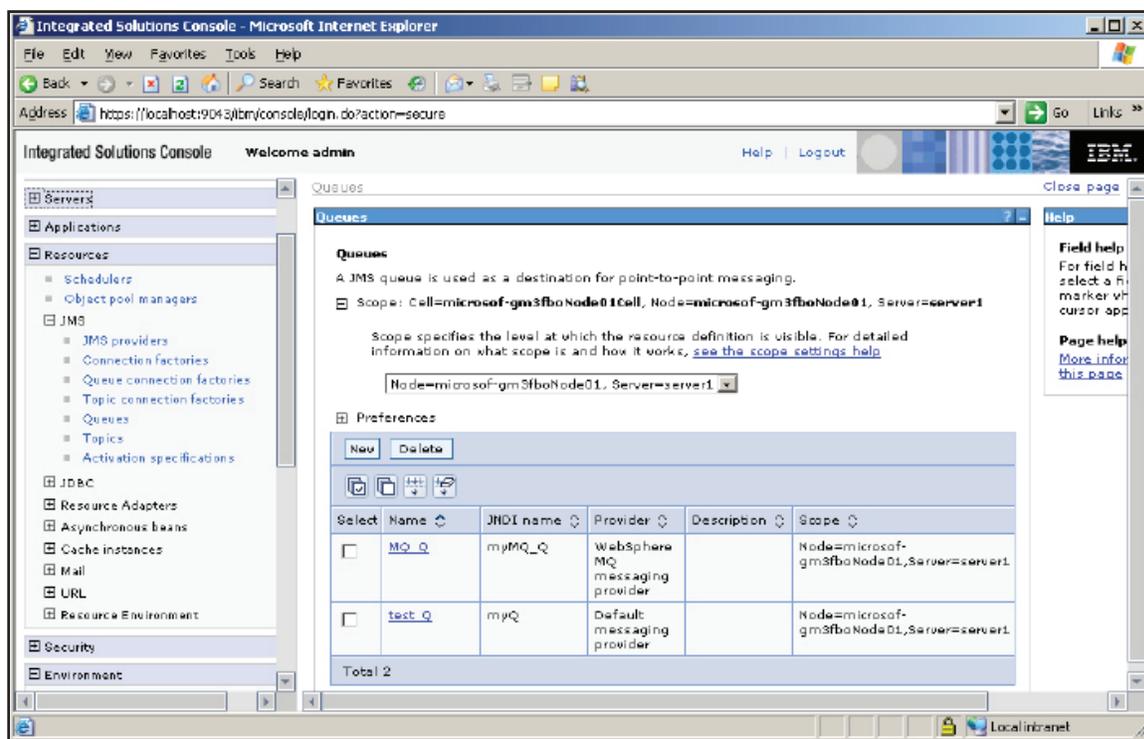


Figure 1. The JNDI tree

Using the RefFSContextFactory JNDI with WebSphere MQ

Stand-alone WebSphere MQ uses a JNDI scaffolding based on the Sun *RefFSContextFactory* implementation. This implements a local JNDI store on each machine where the adapter is installed. Using the WMQ Explorer, the JNDI store can be created under the node, JMS Administered Objects, by using the menu choice, Add Initial Context. The JNDI store is a file, *.bindings*, that can be placed on each BizTalk machine where the adapter is installed.

To correctly use the RefFSContextFactory, please follow these directions.

1. Copy the **.bindings** file from the WMQ machine to the BizTalk machine.
2. Add the JAR file, **jnbjms-fscontext.jar**, to the BizTalk transport handler property, Class Path. This JAR file can be found in the adapter installation support directory in the zip archive, RefFSContextFC.zip.
3. For the BTS transport handler property, Initial Context Factory, use this ICF, **com.sun.jndi.fscontext.JnbBtsRefFSContextFactory**.
4. In the BTS transport handler property, JVM Arguments, point to the *.bindings* file directory (please note the forward strokes):
-Djava.naming.provider.url=file:/C:/[PathToRefFSContextBindingFile]