



**Using the JNBridge JMS Adapter for .NET
with GlassFish and OpenMQ
version 3.0**

www.jnbridge.com

Using the JMS Adapter with GlassFish

JNBridge, LLC
www.jnbridge.com

COPYRIGHT © 2008-2011 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, 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.

Contents

Quick Config for GlassFish/OpenMQ	4
Binding Properties Tab	4
URI Properties Tab	4
Using the JMS Adapter with GlassFish	5
Resources	5
Machine Prerequisites	5
Configuring the Adapter Connection Properties	6
Binding Properties Tab	6
JNBridge Properties Category	7
URI Properties Tab	8
Security Tab	9

Using the JMS Adapter with GlassFish

Quick Config for GlassFish/OpenMQ

Binding Properties Tab

- Initial Context Factory:

For GlassFish 3.x, use this initial context factory:

```
com.sun.enterprise.naming.impl.SerialInitContextFactory
```

For GlassFish 2.1, use this initial context factory:

```
com.sun.appserv.naming.SIASCtxFactory
```

- JMS Scheme: **iiop**

- Queue Factory:

No default connection factories are pre-configured.

- Topic Factory:

No default connection factories are pre-configured.

- Class Path

For GlassFish 3.x, Use this JAR file, **gf-client.jar**. Because this JAR file has an embedded classpath, it must be in situ with the entire GlassFish/OpenMQ directory structure. For GlassFish 2.1, use these JAR files (there is no embedded class path, so use only these):

```
.../glassfish/imq/lib/imq.jar  
.../glassfish/imq/lib/imqutil.jar  
.../glassfish/imq/lib/jms.jar  
.../glassfish/lib/appserv-admin.jar  
.../glassfish/lib/appserv-rt.jar  
.../glassfish/lib/j2ee.jar  
.../glassfish/lib/javaee.jar
```

- JVM Path (example)

```
C:\Program Files\Java\jre1.5.0_10\bin\client\jvm.dll
```

- JVM Arguments

If the GlassFish server is not running on the machine where BizTalk and the adapter are installed, then a JEE system property must be used to specify the machine where the GlassFish server is running. Use the JVM Arguments property with this entry:

```
-Dorg.omg.CORBA.ORBInitialHost=<servername>
```

URI Properties Tab

- Port Number: **3700**

Using the JMS Adapter with GlassFish

This document uses the example JMS service that comes pre-configured in GlassFish/OpenMQ. 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 GlassFish. 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*.
- Chances are, if the target GlassFish implementation is mature, the values for the configuration of BizTalk transport handlers and send/receive ports can be supplied by the GlassFish administrator, gleaned from existing JMS client code or property files, e.g. *jndi.properties*.
- If the GlassFish/OpenMQ implementation targeted is not configured, then the default example JMS broker can be used for proof-of-concept evaluations.
- 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 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 JNBridge JMS Adapter for BizTalk uses the stand-alone JMS environment supplied by SonicMQ. This environment consists of one or more JAR files. Please see the section, *Class Path*, below.

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.

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 *GlassFish* and the default configuration values will automatically appear.

■ 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 GlassFish 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. For GlassFish 3.x, use this initial context factory:

```
com.sun.enterprise.naming.impl.SerialInitContextFactory
```

For GlassFish 2.1, use this initial context factory:

```
com.sun.appserv.naming.SLASCtxFactory
```

Using the JMS Adapter with GlassFish

■ 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 GlassFish, the scheme is:

`iiop`

■ 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 default connection factories are pre-configured.

! *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 default connection factories are pre-configured.

! *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 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.

Using the JMS Adapter with GlassFish

For GlassFish version 3.x, the jar file required by the JMS adapter:

`[GlassFish_Directory]\glassfishv3\glassfish\modules\gf-client.jar`

This JAR file has an embedded classpath in its manifest. As such, it is recommended that the machine where BizTalk and the adapter are installed also have GlassFish installed even if the GlassFish server is running on another machine. The Class Path property must point to the `gf-client.jar` file in situ.

For GlassFish 2.1, use these JAR files (there is no embedded class path, so use only these):

```
.../glassfish/imq/lib/imq.jar
.../glassfish/imq/lib/imqutil.jar
.../glassfish/imq/lib/jms.jar
.../glassfish/lib/appserv-admin.jar
.../glassfish/lib/appserv-rt.jar
.../glassfish/lib/j2ee.jar
.../glassfish/lib/javaee.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 JVM used is:

`C:\Program Files\Java\jre1.6.0_04\bin\client\jvm.dll`

■ JVM Arguments

If the GlassFish server is not running on the machine where BizTalk and the adapter are installed, then a JEE system property must be used to specify the machine where the GlassFish server is running. Use the JVM Arguments property with this entry:

`-Dorg.omg.CORBA.ORBInitialHost=<servername>`

URI Properties Tab

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

■ Host

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

■ Port

This is a text editable field. Enter the port where GlassFish is listening for client connections. By default this is port `3700`.

Security Tab

This tab need only be used if security is implemented in the GlassFish 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