



BEA WebLogic Server®

Configuring and Managing the WebLogic Messaging Bridge

Version 9.0 BETA
Revised: December 15, 2004

BETA

Copyright

Copyright © 2004 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

BETA

Contents

1. Introduction and Roadmap

Document Scope and Audience	1-1
Guide to this Document	1-2
Related Documentation	1-2
Samples and Tutorials for the JMS Developer	1-3
Avitek Medical Records Application (MedRec) and Tutorials	1-3
JMS Examples in the WebLogic Server Distribution	1-3
New and Changed JMS Features in This Release	1-3

2. Understanding the Messaging Bridge

What Is a Messaging Bridge?	2-1
Messaging Bridge Components	2-2
Resource Adapters	2-2
Source and Target Bridge Destinations	2-6
JMS Bridge Destinations	2-7
General Bridge Destinations	2-7
Messaging Bridge Instance	2-8

3. Designing a Messaging Bridge

When to use a Messaging Bridge	3-1
Store-and-Forward Messaging	3-1
Replicating a Topic	3-2
Selecting a Quality-of-Service (QOS) Level	3-2

Messaging Persistence	3-3
Message Ordering	3-3
Setting the Number of ConnectionFactories	3-4
Using Distributed Destinations as Source and Target Destinations	3-4

4. Configuring a Messaging Bridge

Create a Messaging Bridge	4-1
Create a New Destination	4-3
Use an Existing Bridge Destination	4-3
Modify a Messaging Bridge Configuration for Your Environment	4-4
Deploying the Bridge's Resource Adapters	4-4
Configuring Source and Target Bridge Destinations	4-5
Configuring JMS Bridge Destinations	4-5
Configuring General Bridge Destinations	4-7
Modifying an Existing Messaging Bridge Instance	4-9
Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target	4-10

5. Managing a Messaging Bridge

Monitoring All Messaging Bridges	5-1
Stopping and Restarting a Messaging Bridge	5-2
Modifying the Messaging Bridge Execute Thread Pool Size	5-2
Modifying the ConnectionFactory Capacity	5-2

6. Interoperating with Different WebLogic Server Releases

Naming Guidelines for WebLogic Servers and Domains	6-5
Enabling Security Interoperability for WebLogic Domains	6-6
Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain	6-6
Using the Messaging Bridging To Access Destinations In a Release 5.1 Domain	6-7

7. Accessing Third-Party Message Providers

8. Tuning WebLogic Message Bridge

Change the Batch Size	8-1
Change the Batch Interval	8-2
Change the Quality of Service	8-2
Using Multiple Bridge Instances	8-2
Change the Bridge Thread Pool Size	8-3
Avoid Durable Subscriptions	8-3
Co-locate Bridges With Their Source or Target Destination	8-3
Change the Asynchronous Mode Enabled Attribute	8-3
Synchronous Mode	8-4
Asynchronous Mode	8-4

BETA

Introduction and Roadmap

This section describes the contents and organization of this guide—*Configuring and Managing WebLogic Message Bridge*.

- [“Document Scope and Audience” on page 1-1](#)
- [“Guide to this Document” on page 1-2](#)
- [“Related Documentation” on page 1-2](#)
- [“Samples and Tutorials for the JMS Developer” on page 1-3](#)
- [“New and Changed JMS Features in This Release” on page 1-3](#)

Document Scope and Audience

This document is a resource for software developers who want to configure and manage a WebLogic messaging bridge as a forwarding mechanism between any two messaging products—thereby, providing interoperability between separate implementations of WebLogic JMS, or between WebLogic JMS and another messaging product. It also contains information that is useful for business analysts and system architects who are evaluating WebLogic Server® or considering the use of WebLogic Server JMS for a particular application.

It is assumed that the reader is familiar with J2EE and JMS concepts. This document emphasizes the value-added features provided by WebLogic Server and key information about how to use WebLogic Server features and facilities to configure and manage a messaging bridge.

Guide to this Document

- This chapter, [Chapter 1, “Introduction and Roadmap,”](#) introduces the organization of this guide.
- [Chapter 2, “Understanding the Messaging Bridge,”](#) describes basic WebLogic Messaging Bridge resources, such as a resource adaptors and destinations.
- [Chapter 3, “Designing a Messaging Bridge,”](#) explains how to configure messaging bridge features, such as unit of order, persistence, and quality of service.
- [Chapter 4, “Configuring a Messaging Bridge,”](#) describes how configure a messaging bridge, which is a forwarding mechanism between any two messaging products, including separate implementations of WebLogic Server.
- [Chapter 5, “Managing a Messaging Bridge,”](#) explains how to manage the WebLogic Messaging Bridge.
- [Chapter 6, “Interoperating with Different WebLogic Server Releases,”](#) explains the interoperability guidelines that apply when using the messaging bridge to access JMS destinations on different releases of WebLogic Server and in other WebLogic Server domains.
- [Chapter 7, “Accessing Third-Party Message Providers,”](#) provides information on how to integrate WebLogic Server with JMS providers from other vendors.
- [Chapter 8, “Tuning WebLogic Message Bridge,”](#) provides information on how to tune a messaging bridge to achieve improved performance.

Related Documentation

For information on topics related to configuring and managing a messaging bridge, see the following documents:

- [*Configuring and Managing WebLogic JMS*](#) is a guide to configuring and managing WebLogic JMS resources.
- [*Programming WebLogic JMS*](#) is a guide to developing WebLogic JMS applications.
- [*Developing WebLogic Server Applications*](#) is a guide to developing WebLogic Server applications.
- [*Deploying WebLogic Server Applications*](#) is the primary source of information about deploying WebLogic Server applications.

- [WebLogic Server Performance and Tuning](#) contains information on monitoring and improving the performance of WebLogic Server applications.

Samples and Tutorials for the JMS Developer

In addition to this document, BEA Systems provides a variety of code samples and tutorials for JMS developers. The examples and tutorials illustrate WebLogic Server JMS in action, and provide practical instructions on how to perform key JMS development tasks.

BEA recommends that you run some or all of the JMS examples before developing your own EJBs.

Avitek Medical Records Application (MedRec) and Tutorials

MedRec is an end-to-end sample J2EE application shipped with WebLogic Server that simulates an independent, centralized medical record management system. The MedRec application provides a framework for patients, doctors, and administrators to manage patient data using a variety of different clients.

MedRec demonstrates WebLogic Server and J2EE features, and highlights BEA-recommended best practices. MedRec is included in the WebLogic Server distribution, and can be accessed from the Start menu on Windows machines. For Linux and other platforms, you can start MedRec from the `WL_HOME\samples\domains\medrec` directory, where `WL_HOME` is the top-level installation directory for WebLogic Platform.

MedRec includes a service tier comprised primarily of Enterprise Java Beans (EJBs) that work together to process requests from web applications, web services, and workflow applications, and future client applications. The application includes message-driven, stateless session, stateful session, and entity EJBs.

JMS Examples in the WebLogic Server Distribution

WebLogic Server 9.0 optionally installs API code examples in

`WL_HOME\samples\server\examples\src\examples`, where `WL_HOME` is the top-level directory of your WebLogic Server installation. You can start the examples server, and obtain information about the samples and how to run them from the WebLogic Server 9.0 Start menu.

New and Changed JMS Features in This Release

The following sections provides information on new and changed features for this release of the WebLogic messaging bridge:

- Forwarded message IDs are always preserved for WebLogic JMS-to-WebLogic JMS communications.
- The introduction of the `PreserveMsgProperty` configuration parameter, which preserves the following properties for 9.0 target bridge destinations: message ID, priority, expiration time, message timestamp, user ID, delivery mode, priority, expiration time, redelivery limit, and unit-of-order name. However, only the delivery mode, priority, and expiration time will be preserved for pre-release 9.0 target bridge destinations.
- For easier bridge adapter updates, WebLogic Server supports an exploded format of the RAR adapters (`jms-xa-adp`, `jms-notran-adp`, and `jms-notran-adp51`).
- Administration console provides a simplified configuration process.

For release-specific information, see these sections in *WebLogic Server 9.0 Release Notes*:

- [“WebLogic Server 9.0 Features and Changes”](#) lists new, changed, and deprecate features.
- [“WebLogic Server 9.0 Known Issues”](#) lists known problems by service pack, for all WebLogic Server APIs, including WebLogic Messaging Bridge.

Understanding the Messaging Bridge

These sections review the different WebLogic Messaging Bridge concepts and features, and describe how they work with WebLogic Server.

It is assumed the reader is familiar with other WebLogic Server administration concepts.

- [“What Is a Messaging Bridge?” on page 2-1](#)
- [“Messaging Bridge Components” on page 2-2](#)

What Is a Messaging Bridge?

The WebLogic Messaging Bridge allows you to configure a forwarding mechanism between any two messaging products—thereby, providing interoperability between separate implementations of WebLogic JMS, or between WebLogic JMS and another messaging product. You can use the Messaging Bridge to integrate your messaging applications between:

- Any two implementations of WebLogic JMS, including those from separate releases of WebLogic Server.
- WebLogic JMS implementations that reside in separate WebLogic domains.
- WebLogic JMS with a third-party JMS product (for example, MQSeries).
- WebLogic JMS with non-JMS messaging products (only by using specialized connector adapters that are not provided with WebLogic Server).

A messaging bridge instance forwards messages between any two destinations. These destinations are mapped to a pair of bridge source and target destinations. The source bridge

destination is where the messaging bridge reads messages from, while the target bridge destination is where the messaging bridge forwards the messages that it receives from the source bridge destination.

For WebLogic JMS and third-party JMS products, a messaging bridge communicates with source and target destinations using the JCA resource adapters provided with WebLogic Server. For non-JMS messaging products, a custom connector adapter must be obtained from a third-party OEM vendor or by contacting BEA Professional Services in order to access non-JMS source or target destinations.

Source and target bridge destinations can be designated as either queues, topics, or [distributed destinations](#). You can also specify a quality of service (QOS), as well as message filters, transaction semantics, and connection retry policies. Once a messaging bridge is configured, it is easily managed from the Administration Console, including the ability to temporarily suspending bridge traffic whenever necessary, tuning the execute thread pool size to suit your implementation, and monitoring the status of all your configured bridges.

Messaging Bridge Components

This section provides information on resources you need to use a messaging bridge:

- [“Resource Adapters” on page 2-2](#)
- [“Source and Target Bridge Destinations” on page 2-6](#)
- [“Messaging Bridge Instance” on page 2-8](#)

Resource Adapters

A messaging bridge uses J2EE Connector Architecture (JCA) resource adapters to communicate with the configured source and target JMS destinations. You need to associate both the source and target JMS destinations with a *supported* resource adapter in order for the bridge to communicate with them. The JNDI name for the adapter is configured as part of the resource adapter’s deployment descriptor.

Note: Although WebLogic JMS includes a “General Bridge Destination” framework for accessing non-JMS messaging products, WebLogic Server does not provide supported connector adapters for such products. Therefore, you must obtain a custom connector adapter from a third-party OEM vendor and consult their documentation for configuration instructions. You can also contact BEA Professional Services for information about obtaining a custom connector adapter.

Resource adaptors for different types of JMS destinations are provided in exploded format or in a `.rar` file. The exploded format provides you an easy way to modify resource adaptor deployment descriptors, such as the `max-capacity` of the connection factory which specifies the maximum number of connections available for bridge instances. Note, changing a deployment descriptor for a resource adaptor using the exploded format does not update the descriptor packaged in the `.rar` file. See [“Modifying the ConnectionFactory Capacity” on page 5-2](#).

BETA

The supported resource adapters are located in the `WL_HOME\server\lib` directory and are described in the following table.

BETA

Table 2-1 Messaging Bridge Resource Adapters and JNDI Names

Adapter	JNDI Name	Description
jms-xa-adp	eis.jms.WLSConnectionFactoryJNDIXA	<p>Provides transaction semantics via the XAResource. Used when the required QOS is <i>Exactly-once</i>. This envelops a received message and sends it within a user transaction (XA/JTA). The following requirements are necessary in order to use this resource adapter:</p> <ul style="list-style-type: none"> Any WebLogic Server implementation being bridged must be release 6.1 or later. The source and target JMS connection factories must be configured to use the XAConnectionFactory. <p>Note: Before deploying this resource adapter, refer to the “Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain” on page 6-6 for specific transactional configuration requirements and guidelines.</p>

Table 2-1 Messaging Bridge Resource Adapters and JNDI Names

Adapter	JNDI Name	Description
jms-notran-adp	eis.jms.WLSConnectionFactoryJNDINoTX	<p>Provides no transaction semantics. Used when the required QOS is <i>Atmost-once</i> or <i>Duplicate-okay</i>. If the requested QOS is <i>Atmost-once</i>, the resource adapter uses the <code>AUTO_ACKNOWLEDGE</code> mode. If the requested QOS is <i>Duplicate-okay</i>, <code>CLIENT_ACKNOWLEDGE</code> is used.</p> <p>Note: For more information about the acknowledge modes used in non-transacted sessions, see “Understanding WebLogic JMS” in <i>Programming WebLogic JMS</i>.</p>
jms-notran-adp51	eis.jms.WLS51ConnectionFactoryJNDINoTX	<p>Provides interoperability with release 5.1 when either the source or target destination is a 5.1 server instance. This resource adapter provides no transaction semantics; therefore, it only supports a QOS of <i>Atmost-once</i> or <i>Duplicate-okay</i>. If the requested QOS is <i>Atmost-once</i>, the resource adapter uses the <code>AUTO_ACKNOWLEDGE</code> mode. If the requested QOS is <i>Duplicate-okay</i>, <code>CLIENT_ACKNOWLEDGE</code> is used.</p>

Source and Target Bridge Destinations

A messaging bridge connects two actual destinations that are mapped to bridge destinations: a source destination *from which* messages are received, and a target destination *to which* messages are sent. Depending on the messaging products that need to be bridged, there are two types of bridge destinations:

- [“JMS Bridge Destinations”](#) on page 2-7
- [“General Bridge Destinations”](#) on page 2-7

Refer to the [“Interoperating with Different WebLogic Server Releases”](#) on page 6-5 or [“Accessing Third-Party Message Providers”](#) on page 7-1 sections for specific configuration requirements and guidelines.

When configuring third-party JMS provider bridge destination, you can use the Foreign JMS Server feature to quickly configure multiple source or target destinations. See [Simplified Access to Remote or Foreign JMS Providers](#) in *Programming WebLogic JMS*.

JMS Bridge Destinations

For JMS messaging products, whether it is a WebLogic JMS implementation or a third-party JMS provider, you need to configure a `JMSBridgeDestination` instance for each actual source and target JMS destination being mapped to a messaging bridge. A `JMSBridgeDestination` instance defines a unique name for a bridge’s source and target destinations within a WebLogic domain, the name of the adapter used to communicate with the specified destination, property information to pass to the adapter (Connection URL, Connection Factory JNDI Name, etc.), and, optionally, a user name and password. You need to configure a `JMSBridgeDestination` instance for each actual source and target JMS destination to be mapped to a messaging bridge. See [“Configuring a Messaging Bridge”](#) on page 4-1.

General Bridge Destinations

For non-JMS messaging products, you need to configure a generic `BridgeDestination` instance for each actual source and target destination being mapped to a messaging bridge. A general `BridgeDestination` instance defines a unique name for the actual source and target general bridge destinations within the WebLogic domain, the name of the adapter used to communicate with the specified destination, a list of properties to pass to the adapter, and, optionally, a user name and password.

Note: Although WebLogic JMS includes a “General Bridge Destination” framework for accessing non-JMS messaging products, WebLogic Server does not provide supported connector adapters for such products. Therefore, you must obtain a custom connector adapter from a third-party OEM vendor and consult their documentation for configuration instructions. You can also contact BEA Professional Services for information about obtaining a custom connector adapter.

You need to configure a `BridgeDestination` instance for each actual source and target destination to be mapped to a messaging bridge. See [“Configuring a Messaging Bridge”](#) on page 4-1.

Messaging Bridge Instance

A messaging bridge instance communicates with the configured source and target bridge destinations. For each mapping of a source destination to a target destination, whether it is another WebLogic JMS implementation, a third-party JMS provider, or another non-JMS messaging product, you must configure a `MessagingBridge` instance. Each `MessagingBridge` instance defines the source and target destination for the mapping, a message filtering selector, a QOS, transaction semantics, and various reconnection parameters. See [“Configuring a Messaging Bridge” on page 4-1](#).

BETA

Designing a Messaging Bridge

These sections discuss design options and decisions the administrators and developers should consider before configuring a WebLogic messaging bridge.

- [“When to use a Messaging Bridge” on page 3-1](#)
- [“Selecting a Quality-of-Service \(QOS\) Level” on page 3-2](#)
- [“Messaging Persistence” on page 3-3](#)
- [“Message Ordering” on page 3-3](#)
- [Setting the Number of ConnectionFactories](#)
- [“Using Distributed Destinations as Source and Target Destinations” on page 3-4](#)

When to use a Messaging Bridge

The following sections provide information on when to use a messaging bridge:

- [“Store-and-Forward Messaging” on page 3-1](#)
- [“Replicating a Topic” on page 3-2](#)

Store-and-Forward Messaging

A messaging bridge can be used to effect high availability of remote destinations.

Store-and-Forward messaging enables a local client to produce to a local destination and have those messages automatically forwarded to the remote destination when it is available. This

allows a local client to continue to produce messages when a remote destination is not available. Use the WebLogic Messaging Bridge to provide an administrative solution to store and forward messages between:

- Any two implementations of WebLogic JMS, including those from separate releases of WebLogic Server.
- WebLogic JMS implementations that reside in separate WebLogic domains.
- WebLogic JMS with a third-party JMS product (for example, MQSeries).
- WebLogic JMS with non-JMS messaging products (only by using specialized connector adapters that are not provided with WebLogic Server).

The following table provides compares design features of several forwarding technologies:

Table 3-1 Comparing Message Forwarding Technologies

Feature	Messaging Bridge	Message Driven Beans	WebLogic Store-and-Forward
Implementation mechanism	Administrative	Programmatic	Administrative
Support for foreign and legacy providers	Yes	Yes	No, use to forward messages between releases of WebLogic 9.0.

Replicating a Topic

A messaging bridge can be used to replicate a topic, similar to the distributed topics feature available in WebLogic Server releases 8.1 and 7.0, consequently improving scalability and high availability in some scenarios. Topic replication is accomplished by configuring the bridge to subscribe to one topic and forward the topic's messages to another topic, in essence creating two topics with the same message stream.

Selecting a Quality-of-Service (QOS) Level

The Messaging Bridge supports three different QOS levels:

- Exactly-once—the highest QOS, guarantees that a message is sent to the remote endpoint once and only once. With Exactly-once, the messages will survive server crashes and network down time, while guaranteeing one occurrence of each message at the endpoint.
- At-least-once—guarantees that a message is sent to the remote endpoint, but with the possibility of duplicates. With At-least-once, multiple copies of a message might show up on the remote endpoint due to any network failures or server crashes before the message lands at the remote endpoint.
- At-most-once—the lowest QOS, does not guarantee that a message is sent to the endpoint, but it guarantees that each message is only sent to the remote endpoint once, if any. With At-most-once, messages may get lost due to any network failures or server crashes; however, there will never be duplicate messages in the endpoint.

In some instances, the target destination may not be able to provide the quality of service configured for the bridge. In these cases, you can configure the bridge instance to allow the quality of service to be degraded by setting the `QOSDegradationAllowed` flag.

Messaging Persistence

The messaging bridge can save messages that are sent to it while the bridge is not running. The bridge will forward these messages to the target destination when it is restarted. A messaging bridge will store and forward messages to a target destination under the following conditions:

- The source destination is a queue.
- The `Durability Enabled` attribute is set and the source destination is a topic. This creates a durable subscription. For more information, see [Setting Up Durable Subscriptions](#) in *Programming WebLogic JMS*.

Message Ordering

If an application message is in a transaction, saving the message in the persistent storage must be part of the user transaction to preserve exactly-once semantics. In particular, the message will be removed from the persistent storage as part of the transaction rollback if the application decides to rollback the transaction. However, forwarding is NOT part of the application transaction. The sending agent will not forward a transactional message until the transaction commits. Within a transaction, message ordering is preserved based on when the messages are sent. To ensure message ordering, configure a Unit-of-Order. See [Using Message Unit-of-Order](#) in *Programming WebLogic JMS*.

Setting the Number of ConnectionFactories

You may need to modify the capacity of the connection factory associated with each resource adaptor by adjusting the `initial-capacity` and `max-capacity` attributes the `weblogic-ra.xml` descriptor file. In general, the value of the `max-capacity` attribute should be at least two times the number of bridge instances. See [“Modifying the ConnectionFactory Capacity” on page 5-2](#).

For example:

The default configuration sets the value of the `max-capacity` attribute to 20. This setting is adequate for environments that have up to ten message bridge instances targetted. If you increase the number of bridge instances to 15, increase the `max-capacity` attribute to 30.

Using Distributed Destinations as Source and Target Destinations

A messaging bridge can send to and receive from distributed destinations. Bea recommends the following configurations:

- If the source is distributed destination, the bridge is pinned to one of the members when it connects to the destination. It stays connected only to that member until it reconnects. This means that the bridge will not receive messages from the other members of the distributed destination. Therefore, the best practice is to configure one bridge for each member of a distributed destination using the member's JNDIName.
- If the target is a distributed destination, the best practice is to send to the distributed destination using the distributed destination's JNDIName and disable server affinity. This allows the distributed destination to load balance incoming messages.

Configuring a Messaging Bridge

Configuring a messaging bridge instance consists of the following tasks:

1. [“Create a Messaging Bridge” on page 4-1.](#)
2. [“Modify a Messaging Bridge Configuration for Your Environment” on page 4-4.](#)

When you initially create a message bridge, many attributes are set with the default value. You may need to change message bridge settings to better suit your environment. For example, you may wish to select `QOS_Degradation_Allowed` rather than have the messaging bridge use the initially configured QOS.

Create a Messaging Bridge

Creating a messaging bridge consists of the following tasks:

1. Create source and target bridge destinations.
2. Deploy a resource adaptor.
3. Create a messaging bridge instance.
4. Deploy the messaging bridge.

To simplify these tasks, the Administration Console assists you in creating a messaging bridge by deploying an appropriate resource adaptor and setting the values of some attributes. You may need to review the message bridge configuration and modify some attribute settings to better suit your environment. See [“Modify a Messaging Bridge Configuration for Your Environment” on page 4-4](#) for more information on how to modify an existing messaging bridge.

Note: Click the Help button to access detailed information for each panel.

Use the following steps to configure a messaging bridge instance:

1. Click to expand the Services—~~JMS~~—Messaging Bridge nodes.
2. Click Bridges
3. Click New.
4. Modify the configuration attributes for your bridge.
5. Click Next.
6. Configure the source destination.
 - To create a new source destination, follow the instructions at [“Create a New Destination” on page 4-3](#).
 - If you have already configured a source destination, follow the instructions at [“Use an Existing Bridge Destination” on page 4-3](#).
7. Configure the target destination.
 - To create a new source destination, follow the instructions at [“Create a New Destination” on page 4-3](#).
 - If you have already configured a source destination, follow the instructions at [“Use an Existing Bridge Destination” on page 4-3](#).
8. Select the servers and/or clusters on which you want to deploy your new bridge. See [“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 4-10](#) for general information on targeting a messaging bridge.
9. Click Next.

A messaging bridge instance is created. The console checks your configuration and lists any tasks you must perform manually. You may need to perform one or more of the following tasks:

- At least one of the destinations is in a remote Weblogic domain. The “Guest Disabled” check box must not be checked on that domain.
- The “Enable Generated Credential” check box must not be checked. Use the same password as the credential on all Weblogic domains that participate in the bridge.
- The “Enable Generated Credential” check box must not be checked. Use the “system” user’s password as the credential on all Weblogic domains that participate in the bridge.

- The “system” user must be configured in all domains with the same password. The “system” must be a member of the Administrators group in the domain where the bridge resides.
 - The XA connection factory must be enabled for all domains.
 - The jms51-interop.jar file must be in the CLASSPATH of the Weblogic Server 8.1 server instance.
10. Click Finish.
 11. If necessary, perform any tasks displayed in Step 9. See [“Interoperating with Different WebLogic Server Releases” on page 6-5](#) and [“Interoperating with Different WebLogic Server Releases” on page 6-5](#) for additional information on how to resolve these tasks.

Create a New Destination

Use the following instructions to create a new destination for your messaging bridge instance:

1. Click the New JMS Destination button.
1. Modify the configuration attributes for your destination.
2. Click Next.
3. Click the Drop-down arrow to view a list of message providers.
4. Select the message provider for your destination.
5. Click Next.

Use an Existing Bridge Destination

Use the following instructions to configure your messaging bridge instance for an existing destination:

1. Click the Drop-down arrow to view a list of destinations.
2. Select the destination for your messaging bridge.
3. Click Next.
4. Click the Drop-down arrow to view a list of message providers.
5. Select the message provider for your destination.

6. Click Next.

Modify a Messaging Bridge Configuration for Your Environment

The following sections provide information on how to modify the configuration of individual message bridge components:

- [“Deploying the Bridge’s Resource Adapters” on page 4-4](#)
- [“Configuring Source and Target Bridge Destinations” on page 4-5](#)
- [“Modifying an Existing Messaging Bridge Instance” on page 4-9](#)
- [“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 4-10](#)

Deploying the Bridge’s Resource Adapters

Before you configure the messaging bridge destinations, deploy the appropriate resource adapters in the WebLogic Server domain that is hosting the messaging bridge, as follows:

1. Select the domain in which you will deploy the adapters (for example, `wl_server`).
2. Click Deployments.
3. Click Install.
4. Follow the deployment assistant instructions to deploy the appropriate resource adapter located in the `WL_HOME\server\lib` directory, as defined in [Table 2-1, “Messaging Bridge Resource Adapters and JNDI Names,” on page 2-5](#). You can deploy a resource adapter in exploded format or as a `.rar` file.

- `jms-xa-adp`
- `jms-notran-adp`
- `jms-notran-adp51`

Note: When configuring a messaging bridge to interoperate between WebLogic Server release 9.0 and higher and release 5.1, then the release 5.1 resource adapter (`jms-notran-adp51.rar`) and the non-transaction adapter (`jms-notran-adp.rar`) must be deployed on the 9.0 domain running the messaging bridge.

For more information on deploying resource adapters, see [“Packaging and Deploying Resource Adapters”](#) in *Programming WebLogic Resource Adapters*.

Configuring Source and Target Bridge Destinations

A messaging bridge connects two actual destinations that are mapped to bridge destinations: a source destination *from which* messages are received, and a target destination *to which* messages are sent. Depending on the messaging products that need to be bridged, there are two types of bridge destinations:

- **JMS Bridge Destination** – For JMS messaging products, whether it is a WebLogic JMS implementation or a third-party JMS provider, you need to configure a `JMSBridgeDestination` instance for each actual source and target JMS destination being mapped to a messaging bridge.
- **General Bridge Destination** – For non-JMS messaging products, you need to configure a generic `BridgeDestination` instance for each actual source and target destination being mapped to a messaging bridge.

Before starting the procedures in this section, refer to the [“Interoperating with Different WebLogic Server Releases” on page 6-5](#) or [“Accessing Third-Party Message Providers” on page 7-1](#) sections for specific configuration requirements and guidelines.

Note: When configuring third-party JMS provider bridge destination, you can use the Foreign JMS Server feature to quickly configure multiple source or target destinations. For more information, see [“Simplified Access to Remote or Foreign JMS Providers”](#) in *Programming WebLogic JMS*.

Configuring JMS Bridge Destinations

A `JMSBridgeDestination` instance defines a unique name for a bridge’s source and target destinations within a WebLogic domain, the name of the adapter used to communicate with the specified destination, property information to pass to the adapter (Connection URL, Connection Factory JNDI Name, etc.), and, optionally, a user name and password.

You may need to modify or create a `JMSBridgeDestination` instance for each actual source and target JMS destination to be mapped to a messaging bridge. If you are creating destinations for a new bridge instance, you will designate the source and target JMS Bridge Destinations in [“Create a Messaging Bridge” on page 4-1](#). To update destinations used by an existing bridge instances, see [“Modifying an Existing Messaging Bridge Instance” on page 4-9](#).

To configure a source or target JMS bridge destination, follow these steps.

1. Click to expand the Services—~~JMS~~—Messaging nodes.

2. Click the JMS Bridge Destinations node to open the JMS Bridge Destinations page in the right pane.
3. Configure a destination.
 - To create a new destination, click New.
 - To modify an existing destination, click the name of the destination.
4. On the Configuration tab, define the configuration attributes for a JMS bridge destination:
 - Enter a JMS bridge destination name for the actual JMS destination being mapped to the bridge. This name must be unique across a WebLogic domain. For example, if you are bridging between WebLogic Server releases 6.1 and 8.1, for the source destination you could change the default bridge destination name to “61SourceDestination”. Then, when you create the corresponding target destination, you could name it as “81TargetDestination”. Once the bridge destinations are configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges → General tab.
 - Specify the JNDI name of the resource adapter used to communicate with the messaging bridge destinations:
 - `eis.jms.WLSConnectionFactoryJNDIXA` (default) — QOS is *Exactly-once*
 - `eis.jms.WLSConnectionFactoryJNDINoTX` — QOS is *Atmost-once* or *Duplicate-Okay*
 - `eis.jms.WLS51ConnectionFactoryJNDINoTX` — used only with release 5.1 and only supports QOS of *Atmost-once* or *Duplicate-Okay*For more information on which resource adapter name to use, see [“Resource Adapters” on page 2-2](#).
 - Leave the Adapter Classpath field blank when connecting to source and target destinations that are both running on WebLogic Server 6.1 or later. When connecting to either a source or target destination that is running on WebLogic Server 6.0 or earlier, the Adapter Classpath field must indicate the location of the classes for the earlier WebLogic Server release. When connecting to a third-party JMS provider, the bridge destination must supply the provider’s CLASSPATH in the WebLogic Server CLASSPATH.
 - Specify the URL of the JNDI provider used to look up the connection factory and destination.
 - Specify the context factory used to get the JNDI context.
 - Specify the JMS connection factory used to create a connection for the actual JMS destination being mapped to the JMS bridge destination.

- In order to use the Exactly-once QOS for transactions, the JMS connection factory has to be an XA connection factory. Specify the JNDI name of the actual JMS destination being mapped to the JMS bridge destination.
- Indicate whether the destination type is either a Queue or Topic.
- Optionally, enter the user name and password that the messaging bridge will give to the bridge adapter.

Note: All operations done to the specified destination are done using that user name and password. Therefore, the User Name/Password for the source and target destinations must have permission to access the underlying JMS destinations in order for the messaging bridge to work.

5. Click Save to create an instance of the bridge destination with the name you specified in the Name field. The new instance is added under the JMS Bridge Destination node in the left pane.

Configuring General Bridge Destinations

A general `BridgeDestination` instance defines a unique name for the actual source and target general bridge destinations within the WebLogic domain, the name of the adapter used to communicate with the specified destination, a list of properties to pass to the adapter, and, optionally, a user name and password.

Note: Although WebLogic JMS includes a “General Bridge Destination” framework for accessing non-JMS messaging products, WebLogic Server does not provide supported connector adapters for such products. Therefore, you must obtain a custom connector adapter from a third-party OEM vendor and consult their documentation for configuration instructions. You can also contact BEA Professional Services for information about obtaining a custom connector adapter.

You need to configure a `BridgeDestination` instance for each actual source and target destination to be mapped to a messaging bridge. Therefore, when you finish defining attributes for a source general bridge destination, repeat these steps to configure a target general bridge destination, or vice versa. You will designate the source and target general Bridge Destinations in [“Modifying an Existing Messaging Bridge Instance” on page 4-9](#).

To configure a source or target general bridge destination, follow these steps.

1. Expand the Services → Messaging Bridge node.
2. Click the General Bridge Destinations node to open the General Bridge Destinations page in the right pane.

3. Click the Configure a new General Bridge Destination link. A Configuration dialog shows the tabs associated with configuring a new general bridge destination.
4. On the Configuration General tab, define the general configuration attributes for a general bridge destination:
 - Enter a general bridge destination name that is unique across a WebLogic Server domain. For example, if you are bridging between WebLogic Server releases 6.1 and 8.1, for the source destination you could change the default destination name to “61SourceDestination”. Then when you create the corresponding target destination, you could name it as “81TargetDestination”. Once the bridge destinations are configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges →General tab.

- Specify the JNDI name of the adapter used to communicate with the bridge destinations.

`eis.jms.WLSConnectionFactoryJNDIXA` (default) — QOS is *Exactly-once*

`eis.jms.WLSConnectionFactoryJNDINoTX` — QOS is *Atmost-once* or *Duplicate-Okay*

`eis.jms.WLS51ConnectionFactoryJNDINoTX` — used only with release 5.1 and only supports QOS of *Atmost-once* or *Duplicate-Okay*

For more information on which resource adapter name to use, see [“Resource Adapters” on page 2-2](#).

Note: WebLogic Server does not provide adapters for non-JMS messaging products. Therefore, you must use a specialized adapter from a third-party OEM vendor, or contact BEA Professional Services to obtain a custom adapter.

- Leave the Adapter Classpath field blank when connecting to source and target destinations that are both running on WebLogic Server 6.1 or later. When connecting to either a source or target destination that is running on WebLogic Server 6.0 or earlier, the Adapter Classpath field must indicate the location of the classes for the earlier WebLogic Server release. When connecting to a third-party JMS provider, the bridge destination must supply the provider’s CLASSPATH in the WebLogic Server CLASSPATH.

Note: Specify all the properties defined for a bridge destination. Each property must be separated by a semicolon (for example, `DestinationJNDIName=myTopic;DestinationType=topic;`). For non-JMS messaging products that use adapters provided by a third-party OEM vendor, you should consult the vendor’s documentation for property configuration instructions.

- Optionally, enter the user name and password that the messaging bridge will give to the bridge adapter.

Note: The User Name/Password for the source and target destinations must have permission to access the underlying source and target destinations in order for the Messaging Bridge to work.

5. Click Create to create an instance of the general bridge destination with the name you specified in the Name field. The new instance is added under the General Bridge Destination node in the left pane.

Modifying an Existing Messaging Bridge Instance

Note: Before starting the procedure in this section, refer to the [“Interoperating with Different WebLogic Server Releases” on page 6-5](#) or [“Accessing Third-Party Message Providers” on page 7-1](#) sections for specific configuration requirements and guidelines.

To modify a messaging bridge, follow these steps:

1. Expand the Services →Messaging Bridge node.
2. Expand the Bridges node to open the Messaging Bridges tab in the right pane.
3. Click the messaging bridge that you want to modify. A dialog shows the tabs associated with configuring a new messaging bridge.
4. On the Configuration General tab, define the general configuration attributes for a messaging bridge.
5. Click Save. If you selected the Started check box, the bridge will be in a running state once it is targeted.
6. Optionally, on the Target and Deploy tab, select an independent server instance, a cluster, or a migratable server target on which to deploy the messaging bridge. This must be the same target where the bridge’s resource adapter was deployed. You can also reconfigure deployment targets later if you wish.

For more information, see [“Configuring Source and Target Bridge Destinations” on page 4-5](#).

7. Optionally, on the Connection Retry tab, change the attribute fields or accept the default values as assigned. Since the source and target destinations for a messaging bridge will not always be available, the messaging bridge must be able to reconnect to the destination at some periodic interval. These attributes govern the time between reconnection attempts. Then, click Save.
8. Optionally, on the Transactions tab, change the attribute fields or accept the default values as assigned. Then, click Save.

Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target

You can choose the servers, clusters, or migratable targets in your domain on which you would like to deploy a messaging bridge. You can also reconfigure deployment targets later if you wish.

1. Expand the Messaging Bridge → Bridges node to show the list of messaging bridges defined in your domain.
 2. Click the messaging bridge that you want to assign to a server, cluster, or migratable target. A dialog displays in the right pane showing the tabs associated with the messaging bridge instance.
 3. Click the Target and Deploy tab to display the following targeting options.
 - Independent Servers—you can select a server or servers where the messaging bridge will be deployed. The messaging bridge will be available on all the selected servers.
 - Clusters—you can select a cluster where the messaging bridge will be deployed. The messaging bridge will be available on all servers in the selected cluster. You can also target an individual server or servers within a cluster.
 - Migratable Targets—you can select a WebLogic Server migratable target where the messaging bridge will be deployed. When WebLogic Server is first booted, the messaging bridge is initially available only on the user-preferred server. After that, the bridge can be migrated to another server listed in the migratable target.
- Note:** This must be the same target where the bridge's resource adapter was deployed. For more information, see [“Deploying the Bridge's Resource Adapters” on page 4-4](#).
4. Click Save.

Managing a Messaging Bridge

The following sections explain how to manage the WebLogic Message Bridge from the Administration Console:

- [“Monitoring All Messaging Bridges” on page 5-1](#)
- [“Stopping and Restarting a Messaging Bridge” on page 5-2](#)
- [“Modifying the Messaging Bridge Execute Thread Pool Size” on page 5-2](#)
- [“Modifying the ConnectionFactory Capacity” on page 5-2](#)

Monitoring All Messaging Bridges

To monitor the status of all configured messaging bridges in your domain:

1. Expand the Server node.
2. Select the server instance where the messaging bridges are configured. A dialog displays in the right pane showing the tabs associated with the selected server instance.
3. Select the Services → Bridge tab.
4. Click the Monitor all Messaging Bridge Runtimes text link.
5. A table displays showing all the messaging bridge instances for the server and their status (either as running or not running).

Stopping and Restarting a Messaging Bridge

To temporarily suspend and restart an active messaging bridge:

1. Expand the Messaging Bridge node.
2. Select the messaging bridge instance that you want to suspend.
3. On the Configuration → General tab, clear the Started check box to suspend the bridge.
4. To restart the bridge, select the Started check box.

Modifying the Messaging Bridge Execute Thread Pool Size

You can configure the default execute thread pool size for your messaging bridges. For example, you may want to increase or decrease the default size to reduce competition from the WebLogic Server default thread pool. Entering a value of -1 disables this thread pool and forces a messaging bridge to use the WebLogic Server default thread pool.

1. Click to expand the Environments → Servers nodes.
2. Select the specific server instance where the messaging bridge is configured.
3. In the Configuration tab, select Services.
4. In the Messaging Bridge Configurations section, enter a new value in the Messaging Bridge Thread Pool Size field.
5. Click Save.

Modifying the ConnectionFactory Capacity

You can adjust the capacity of the connection factory associated with each resource adaptor by adjusting the `initial-capacity` and `max-capacity` attributes the `weblogic-ra.xml` descriptor file. Use the following steps to modify the the `weblogic-ra.xml` descriptor file:

1. Using the editor of your choice, update the attribute with the desired value. See [Listing 5-1](#).
2. Deploy the updated adapter.
3. Stop and restart any bridge instance that requires the new values.

Listing 5-1 Example weblogic-ra.xml Descriptor File

```
.  
.   
.   
  
<weblogic-connection-factory-dd>  
  
    <connection-factory-name>WLSJMSConnectionFactoryLocal</connection-fac  
tory-name>  
    <jndi-name>eis/jms/WLSConnectionFactoryJNDILocal</jndi-name>  
        <pool-params>  
            <initial-capacity>0</initial-capacity>  
            <max-capacity>20</max-capacity>  
        </pool-params>  
  
</weblogic-connection-factory-dd>
```

BETA

Interoperating with Different WebLogic Server Releases

The following interoperability guidelines apply when using the messaging bridge to access JMS destinations on different releases of WebLogic Server and in other WebLogic Server domains.

- [“Naming Guidelines for WebLogic Servers and Domains” on page 6-5](#)
- [“Enabling Security Interoperability for WebLogic Domains” on page 6-6](#)
- [“Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain” on page 6-6](#)
- [“Using the Messaging Bridging To Access Destinations In a Release 5.1 Domain” on page 6-7](#)

Note: When the messaging bridge is used to communicate between two domains running different releases of Weblogic Server, a best-practice recommendation is for the messaging bridge to be configured to run on the domain using the latest release of Weblogic Server.

Naming Guidelines for WebLogic Servers and Domains

Unique naming rules apply to all WebLogic Server deployments if more than one domain is involved. Therefore, make sure that:

- WebLogic Server instances and domain names are unique.
- WebLogic JMS server names are unique name across domains.

- If a JMS file store is being used for persistent messages, the JMS file store name must be unique across domains.

Enabling Security Interoperability for WebLogic Domains

Whenever the Exactly-once QOS (quality of service) is required for transactionally transferring messages across different WebLogic Server domains, you must establish a *trust relationship* between these domains by using a common security credential. This requirement applies to the source and target destination domains, as well as to the messaging bridge domain – if the messaging bridge is not running in the same domain as the source or target domains.

Follow these steps to establish a trusted relationship between release 6.1 or later WebLogic domains.

1. Configure the security for the 9.0 domain where the messaging bridge is running as follows:
 - a. Click the Domains node (for example, Examples).
 - b. Select the Security General →Advanced tab.
 - c. If necessary, clear the Enable Generated Credential check box.
 - d. Enter a password for the domain in the Credential field. This password must match the password used for the domain that you are interoperating with.

Note: When interoperating with a release 6.1 domain, the 7.0, 8.1, or 9.0 Credential password must *exactly match* the “system” user password configured for the 6.1 domain.
 - e. Confirm the password in the Confirm field, and then click Apply.
2. When interoperating with a release 6.1 domain, make sure that “system” user is a member of the Administrators group in the 7.0, 8.1, or 9.0 domain.

Note: For more information about WebLogic Server 7.0 or later domain interoperability security, see “[Enabling Trust Between WebLogic Domains](#)” in *Managing WebLogic Security*.

Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain

Use these guidelines when configuring a messaging bridge on a release 9.0 domain to provide “Exactly-once” transactional message communication between two release 6.1 or later domains.

Note: The *Exactly-once* quality of service for two-phase transactions is only supported for release 6.1 or later.

- If a JMS file store is being used for persistent messages, the JMS file store name must be unique across WebLogic domains, as described in [“Naming Guidelines for WebLogic Servers and Domains” on page 6-5](#).
- Make sure that security interoperability between the domains is correctly configured, as described in [“Enabling Security Interoperability for WebLogic Domains” on page 6-6](#).
- Make sure that the XA connection factory is enabled for the domains by selecting the XAConnection Factory Enabled check box on the Services →JMS →JMS Modules → Connection Factories →Configuration →Transactions tab.
- Deploy the transaction resource adapter, `jms-xa-adp.rar`, on the 9.0 domain where the messaging bridge is running, as described in [“Deploying the Bridge’s Resource Adapters” on page 4-4](#).
- When configuring the JMS bridge destinations, as described in [“Configuring Source and Target Bridge Destinations” on page 4-5](#), do the following for both the source and target destinations:
 - In the Adapter JNDI Name field, identify the transaction adapter’s JNDI name, `eis.jms.WLSConnectionFactoryJNDIXA`.
 - Do not enter anything in the Adapter Classpath field.
- On the Configuration →General tab, select a Quality Of Service of *Exactly-once*, as described in [“Modifying an Existing Messaging Bridge Instance” on page 4-9](#).

Using the Messaging Bridging To Access Destinations In a Release 5.1 Domain

When configuring a messaging bridge involves interoperability between WebLogic Server 8.1 and release 5.1, you must configure the following on the Weblogic Server 8.1 implementation that the bridge is running on:

Note: The *Exactly-once* QOS for transactions is not supported for WebLogic Server 5.1.

- The `jms51-interop.jar` file in the `WL_HOME\server\lib` directory must be in the CLASSPATH of the WebLogic Server 8.1 implementation.

- The release 5.1 resource adapter (`jms-notran-adp51.rar`) and the non-transaction adapter (`jms-notran-adp.rar`) must be deployed on the 8.1 bridge domain, as described in [“Deploying the Bridge’s Resource Adapters” on page 4-4](#).
- When configuring the JMS source and target destinations, as described in [“Configuring Source and Target Bridge Destinations” on page 4-5](#), do the following:

In the Adapter JNDI Name field:

- For the 8.1 destination, specify the non-transaction adapter’s JNDI name as `eis.jms.WLSConnectionFactoryJNDINoTX`.
- For the 5.1 destination, specify the 5.1 adapter’s JNDI name as `eis.jms.WLS51ConnectionFactoryJNDINoTX`.

In the Adapter Classpath field:

- For the 8.1 destination, leave the field blank.
- For the 5.1 destination, indicate the location of the classes for the WebLogic Server 5.1 release, as well as the location of the `jms51-interop.jar` file for the 8.1 release.

For example, if you have WebLogic Server 5.1 GA installed in a directory named `WL51_HOME` and your WebLogic Server 8.1 release is installed in `WL81_HOME`, then set the Adapter Classpath as follows for the 5.1 destination:

```
WL51_HOME\classes;WL51_HOME\lib\weblogicaux.jar;  
WL81_HOME\server\lib\jms51-interop.jar
```

Note: If your implementation is using a 5.1 Service Pack, the corresponding `sp.jar` files must also be added to the Adapter Classpath field.

- On the Messaging Bridge → Configuration → General tab, select a Quality Of Service of *Atmost-once* or *Duplicate-okay*, as described in [“Modifying an Existing Messaging Bridge Instance” on page 4-9](#).

Accessing Third-Party Message Providers

When configuring a messaging bridge involves interoperability with a third-party messaging provider, you must configure the following:

- Before starting WebLogic Server:
 - Supply the provider's `CLASSPATH` in the WebLogic Server `CLASSPATH`.
 - Include the `PATH` of any native code required by the provider's client-side libraries in the WebLogic Server system `PATH`. (This variable may vary depending on your operating system.)
- In the `JMSBridgeDestination` instance for the third-party messaging product being bridged, provide *vendor-specific* information in the following attributes:
 - Connection URL
 - Initial Context Factory
 - Connection Factory JNDI Name
 - Destination JNDI Name

Note: The messaging bridge cannot provide the “Exactly-once” quality of service when the source and target bridge destinations are located on the same resource manager (that is, when the bridge is forwarding a global transaction that is using the XA resource of the resource manager). For example, when using MQ Series, it is not possible to use the same Queue Manager for the source and target bridge destinations.

For more information on configuring the remaining attributes for a JMS Bridge Destination, see [“Configuring JMS Bridge Destinations” on page 4-5](#).

BETA

Tuning WebLogic Message Bridge

The main objective when tuning a messaging bridge is to improve overall messaging performance. Raw speed, though important, is only one of several performance-related factors. Other performance factors include reliability, scalability, manageability, monitoring, user transactions, message-driven bean support, and integration with an application server.

The following sections provide information on various methods to improve message bridge performance:

- [“Change the Batch Size” on page 8-1](#)
- [“Change the Batch Interval” on page 8-2](#)
- [“Change the Quality of Service” on page 8-2](#)
- [“Using Multiple Bridge Instances” on page 8-2](#)
- [“Change the Bridge Thread Pool Size” on page 8-3](#)
- [“Avoid Durable Subscriptions” on page 8-3](#)
- [“Co-locate Bridges With Their Source or Target Destination” on page 8-3](#)
- [“Change the Asynchronous Mode Enabled Attribute” on page 8-3](#)

Change the Batch Size

When the `Asynchronous Mode Enabled` attribute is set to false and the quality of service is `Exactly-once`, the `Batch Size` attribute can be used to reduce the number of transaction

commits by increasing the number of messages per transaction (batch). The best batch size for a bridge instance depends on the combination of JMS providers used, the hardware, operating system, and other factors in the application environment.

Change the Batch Interval

When the `Asynchronous Mode Enabled` attribute is set to false and the quality of service is `Exactly-once`, the `BatchInterval` attribute is used to adjust the amount of time the bridge waits for each batch to fill before forwarding batched messages. The best batch interval for a bridge instance depends on the combination of JMS providers used, the hardware, operating system, and other factors in the application environment. For example, if the queue is not very busy, the bridge may frequently stop forwarding in order to wait batches to fill, indicating the need to reduce the value of the `BatchInterval` attribute.

Change the Quality of Service

An `Exactly-once` quality of service may perform significantly better or worse than `At-most-once` and `Duplicate-okay`.

When the `Exactly-once` quality of service is used, the bridge must undergo a two-phase commit with both JMS servers in order to ensure the transaction semantics and this operation can be very expensive. However, unlike the other qualities of service, the bridge can batch multiple operations together using `Exactly-once` service.

You may need to experiment with this parameter to get the best possible performance. For example, if the queue is not very busy or if non-persistent messages are used, `Exactly-once` batching may be of little benefit.

Using Multiple Bridge Instances

If message ordering is not required, consider deploying multiple bridges.

Multiple instances of the bridge may be deployed using the same destinations. When this is done, each instance of the bridge runs in parallel and message throughput may improve. If multiple bridge instances are used, messages will not be forwarded in the same order they had in the source destination.

The use of multiple bridges to improve throughput depends on several factors:

- Some JMS products don't seem to benefit much from using multiple bridges

- WebLogic JMS messaging performance typically improves significantly, especially when handling persistent messages.
- Resource saturation. If the CPU or disk storage are already saturated, increasing the number of bridge instances may decrease throughput.

Change the Bridge Thread Pool Size

If more than five synchronous bridges are targeted to the same WebLogic server, increase the size of the bridge thread pool to match the number of bridges instances.

To avoid competing with the default execute thread pool in the server, messaging bridges share a separate thread pool. This thread pool is used only in synchronous mode ([Asynchronous Mode Enabled](#) is not set). In asynchronous mode the bridge runs in a thread created by the JMS provider for the source destination.

Avoid Durable Subscriptions

If the bridge is listening on a topic and it is acceptable that messages are lost when the bridge is not booted, the [Durability Enabled](#) flag should be disabled to ensure undelivered messages don't accumulate in the source server's store. Disabling the flag will also have the effect of making the messages non-persistent.

Co-locate Bridges With Their Source or Target Destination

If a messaging bridge's source or target is a WebLogic destination, deploy the bridge to the same WebLogic server as the destination. Targeting a messaging bridge with one of its destinations eliminates associated network and serialization overhead. Such overhead can be significant in high-throughput applications, particularly if the messages are non-persistent.

Change the Asynchronous Mode Enabled Attribute

The [Asynchronous Mode Enabled](#) attribute determines whether the messaging bridge receives messages asynchronously using the JMS [MessageListener interface](#), or whether the bridge receives messages using the synchronous JMS APIs. In most situations, the [Asynchronous Enabled](#) attributes value is dependent on the QOS required for the application environment as shown in [Table 8-1](#):

Table 8-1 Asynchronous Mode Enabled Values for QOS Level

QOS	Asynchronous Mode Enabled Attribute value
Exactly-once ¹	false
At-least-once	true
At-most-once	true

1. If the source destination is a non-WebLogic JMS provider and the QOS is Exactly-once, then the Asynchronous Mode Enabled attribute is disabled and the messages are processed in asynchronous mode.

Synchronous Mode

A quality of service of [Exactly-once](#) has a significant effect on bridge performance. The bridge starts a new transaction for each message and performs a two-phase commit across both JMS servers involved in the transaction. Since the two-phase commit is usually the most expensive part of the bridge transaction, as the number of messages being processed increases, the bridge performance will tend to decrease.

Asynchronous Mode

In situations where a quality of service of [Exactly-once](#) is not required, the source destination can pipeline or process multiple messages in each transaction. If an application uses asynchronous consumers (such as MDBs), consider increasing the WebLogic JMS Connection Factory's configured [MessagesMaximum](#) value.

Note: Some non-WebLogic JMS implementations do not pipeline messages, or optimize the asynchronous listeners, in which case the `Asynchronous Mode Enabled` parameter may have little effect.

If the asynchronous pipeline size is already set to 1, this may indicate that the application has enabled ordered redelivery, which in turn means that the pipeline must not be increased. See “[Ordered Redelivery of Messages](#)” in *Programming WebLogic JMS*.

WebLogic JMS pipelines messages that are delivered to asynchronous consumers, otherwise known as message listeners. This action aids performance as messages are aggregated when they are internally pushed from the server to the client. The messages backlog (the size of the pipeline) between the JMS server and the client is tunable. You can tune it by configuring the

`MessagesMaximum` setting on the connection factory. See “[Asynchronous Message Pipeline](#)” in *Programming WebLogic JMS*.

In some circumstances, tuning the `MessagesMaximum` parameter may improve performance dramatically. One situation is when the JMS application defers acknowledges or commits. In this case, BEA suggests setting the value to:

$$2 * (\text{ack or commit interval}) + 1$$

For example:

If the JMS application acknowledges 50 messages at a time, set the `MessagesMaximum` value to 101.

Tuning the `MessagesMaximum` value too high can cause:

- Increased memory usage on the client
- Affinity to an existing client as its pipeline fills with messages. For example: If `MessagesMaximum` has a value of 10,000,000, the first consumer client to connect will get all messages that have already arrived at the destination. This condition leaves other consumers without any messages and creates an unnecessary backlog of messages in the first consumer that may cause the system to run out of memory.

BETA