



BEA WebLogic Server™

BEA WebLogic Server Frequently Asked Questions

Copyright

Copyright © 2002 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 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.

BEA WebLogic Server Frequently Asked Questions

Part Number	Document Revised	Software Version
N/A	September 5, 2002	BEA WebLogic Server Version 7.0

Contents

About This Document

Audience..... viii
e-docs Web Site..... viii
How to Print the Document..... ix
Related Information..... ix
Contact Us!..... ix
Documentation Conventionsx

-
- 1. FAQs: Administration and Configuration**
 - 2. FAQs: Applets**
 - 3. FAQs: Clustering**
 - 4. FAQs: Code Examples**
 - 5. FAQs: Deployment**
 - 6. FAQs: EJB**
 - 7. FAQs: Installation**
 - 8. FAQs: Java**
 - 9. FAQs: J2EE Connector Architecture**
 - 10. FAQs: WebLogic JDBC**
 - 11. FAQs: WebLogic jDriver for MSSQL Server**
 - 12. FAQs: WebLogic jDriver for Oracle**
 - 13. FAQs: JMS**
 - 1. FAQs: WebLogic Messaging Bridge**
 - 2. FAQs: JTA**
 - 3. FAQs: Plug-Ins**
 - 4. FAQs: Server-Related Questions**
 - 5. FAQs: Server-Side Java (Servlets)**
 - 6. FAQs: Security**
 - 7. FAQs: Upgrading**

8. FAQs: Web Services

9. FAQs: Wireless-Related Questions

10. FAQs: XML



About This Document

This document introduces BEA WebLogic Server™ Frequently Asked Questions.

Note: Some of the information in this document may be inconsistent with what you observe in the WebLogic Server 7.0 Beta release.

The document is organized as follows:

- [Chapter 1, “FAQs: Administration and Configuration,”](#) contains questions and answers about administering and configuring WebLogic Server.
- [Chapter 2, “FAQs: Applets,”](#) contains questions and answers about applets.
- [Chapter 3, “FAQs: Clustering,”](#) contains questions and answers about clustering.
- [Chapter 4, “FAQs: Code Examples,”](#) contains questions and answers about code examples.
- [Chapter 6, “FAQs: EJB,”](#) contains questions and answers about EJB.
- [Chapter 7, “FAQs: Installation,”](#) contains questions and answers about installing WebLogic Server.
- [Chapter 8, “FAQs: Java,”](#) contains questions and answers about Java.
- [Chapter 9, “FAQs: J2EE Connector Architecture,”](#) contains questions and answers about J2EE Connector Architecture.
- [Chapter 10, “FAQs: WebLogic JDBC,”](#) contains questions and answers about JDBC.
- [Chapter 11, “FAQs: WebLogic jDriver for MSSQL Server,”](#) contains questions and answers about WebLogic jDriver for MSSQL Server.
- [Chapter 12, “FAQs: WebLogic jDriver for Oracle,”](#) contains questions and answers about WebLogic jDriver for Oracle.

-
- [Chapter 13, “FAQs: JMS,”](#) contains questions and answers about JMS.
 - [Chapter 2, “FAQs: JTA,”](#) contains questions and answers about JTA.
 - [Chapter 4, “FAQs: Server-Related Questions,”](#) contains general questions and answers about WebLogic Server.
 - [Chapter 5, “FAQs: Server-Side Java \(Servlets\),”](#) contains questions and answers about servlets.
 - [Chapter 6, “FAQs: Security,”](#) contains questions and answers about security.
 - [Chapter 9, “FAQs: Wireless-Related Questions,”](#) contains questions and answers about wireless technology.
 - [Chapter 10, “FAQs: XML,”](#) contains questions and answers about XML.

Audience

This document is written for application developers who want to build e-commerce applications using the Java 2 Platform, Enterprise Edition (J2EE) from Sun Microsystems. It is assumed that readers know Web technologies, object-oriented programming techniques, and the Java programming language.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation.

How to Print the Document

You can print a copy of this document from a Web browser, one main topic at a time, by using the File→Print option on your Web browser.

A PDF version of this document is available on the WebLogic Server documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Server documentation Home page, click Download Documentation, and select the document you want to print.

Adobe Acrobat Reader is available at no charge from the Adobe Web site at <http://www.adobe.com>.

Related Information

The BEA corporate Web site provides all documentation for WebLogic Server.

Contact Us!

Your feedback on BEA documentation is important to us. Send us e-mail at docsupport@bea.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the documentation.

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at <http://www.bea.com>. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Usage
Ctrl+Tab	Keys you press simultaneously.
<i>italics</i>	Emphasis and book titles.
monospace text	Code samples, commands and their options, Java classes, data types, directories, and file names and their extensions. Monospace text also indicates text that you enter from the keyboard. <i>Examples:</i> <pre>import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float</pre>
<i>monospace italic text</i>	Variables in code. <i>Example:</i> <pre>String CustomerName;</pre>

Convention	Usage
UPPERCASE TEXT	Device names, environment variables, and logical operators. <i>Examples:</i> LPT1 BEA_HOME OR
{ }	A set of choices in a syntax line.
[]	Optional items in a syntax line. <i>Example:</i> <pre>java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]</pre>
	Separates mutually exclusive choices in a syntax line. <i>Example:</i> <pre>java weblogic.deploy [list deploy undeploy update] password {application} {source}</pre>
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> ■ An argument can be repeated several times in the command line. ■ The statement omits additional optional arguments. ■ You can enter additional parameters, values, or other information
.	Indicates the omission of items from a code example or from a syntax line.



1 FAQs: Administration and Configuration

- How do I bypass the prompt for username and password when I start a server?
- Can I start a Managed Server if the Administration Server is unavailable?
- How can I start a server with a clean (default) configuration?
- What is the easiest way to set the classpath?
- What are safeguards for modifying the config.xml file?
- How do I edit the config.xml file?

Q. How do I bypass the prompt for username and password when I start a server?

A. You can create and use a boot identify file, which contains your username and password in an encrypted format.

If a server's root directory contains a valid boot identity file named `boot.properties`, it uses this file by default. If you want to specify a different file (or if you do not want to store boot identity files in a server's root directory), you can specify a file name in the server's startup command. If a server is unable to access its boot identity file, it displays the standard username and password prompt in its command shell and writes a message to the log file.

For more information, refer to [Bypassing the Interactive Prompt](#) in the *WebLogic Server Administration Guide*.

Q. Can I start a Managed Server if the Administration Server is unavailable?

A. By default, if a Managed Server is unable to connect to the specified Administration Server during startup, it can retrieve its configuration by reading a configuration file and other files directly. You cannot change the server's configuration until the Administration Server is available. A Managed Server that starts in this way is running in Managed Server Independence mode. For more information, refer to [Starting a Managed Server When the Administration Server Is Not Accessible](#) in the Managing and Monitoring WebLogic Server Domains guide.

Q. How can I start a server with a clean (default) configuration?

A. If you run into problems in your environments and want to boot a server with a clean (default) configuration, you can start a WebLogic Server in such a way that it generates a new `config.xml` file. This new `config.xml` file contains only the default settings, unless you use command-line arguments to override the defaults. For more information, refer to [Using the Default Configuration to Start a Server](#) in the *WebLogic Server Administration Guide*.

Q. What is the easiest way to set the classpath?

WebLogic Server installs the following script that you can use to set the classpath that a server requires:

`WL_HOME\server\bin\setWLSEnv.cmd` (on Windows)

`WL_HOME/server/bin/setWLSEnv.sh` (on UNIX)

where `WL_HOME` is the directory in which you installed WebLogic Server. For more information, refer to [Setting the Classpath](#) in the *WebLogic Server Administration Guide*.

Q. What are safeguards for modifying the `config.xml` file?

A. BEA recommends that you use the WebLogic Server Administration Console to edit the `config.xml` file. If you need to edit the `config.xml` file directly, we recommend that:

- You always save a copy of your `config.xml` file before editing it. If your edits cause errors that prevent you from booting, you can at least revert to your saved version.
- You manually edit the `config.xml` file for a domain when a domain is not active. If you edit the configuration file while the domain is active, any changes you make manually will likely be overwritten by the system. Furthermore, all changes you make by hand while the domain is active are ignored by the system at run time.

-
- Because there is no validation or value checking when you edit `config.xml` manually, type-checking occurs when you load the configuration file for the first time, that is, when you restart the Administration Server. At that point, any invalid XML or attribute value is detected and the domain fails to boot.

For more information on modifying the `config.xml` file and a reference of Mbeans and attributes used `config.xml`, see the [BEA WebLogic Server Configuration Reference](http://e-docs.bea.com/wls/docs70/config_xml/index.html) at http://e-docs.bea.com/wls/docs70/config_xml/index.html.

Q. How do I edit the `config.xml` file?

A. The persistent configuration for a domain of WebLogic Servers and clusters is stored in an XML configuration file (`config.xml`). You can modify this file in the following ways:

- The recommended way for users to modify or monitor a domain's configuration is to use the Administration Console.
- WebLogic Server includes a configuration Application Programming Interface (API), which programs can use to modify configuration attributes of resources in the domain.
- You can access the attributes of a domain with the WebLogic Server command-line utility. This utility is provided for people who want to create scripts to automate domain management. See [Commands for Managing a WebLogic Domain](#) in the *Administration Guide* at <http://e-docs.bea.com/wls/docs70/adminguide/cli.html>.
- If you intend to edit the `config.xml` file directly, (not recommended) see the [BEA WebLogic Server Configuration Reference](#) at http://e-docs.bea.com/wls/docs70/config_xml/index.html.

2 FAQs: Applets

- What are the alternatives to using applets?
- Can I use a “native” two-tier driver for a browser applet?
- Why doesn’t my browser applet connect to the database?
- Why does my applet work with Appletviewer, but not with a browser?
- What’s causing ClassFormatErrors with my applet?

Q. What are the alternatives to using applets?

A. BEA supports the use of server-side applications with HTTP servlets and Java Server Pages (JSPs) as part of the J2EE platform. We recommend that before you develop new applications, consider using either servlets or JSPs. A well-designed series of interactive Web pages using servlets and Java Server Pages (JSPs) generally yield a faster and more reliable Web site. If you are currently using applets, you may find that most can be converted to Java applications using Java Web Start and you can continue using WebLogic Server. For information, go to Sun's Java Web Start site.

For more information, see [Using Applets With WebLogic Server](http://e-docs.bea.com/wls/docs70/applets/usingapplets.html) at <http://e-docs.bea.com/wls/docs70/applets/usingapplets.html>.

Q. Can I use a “native” two-tier driver for a browser applet?

A. No. Within an unsigned applet, you cannot load native libraries over the wire, access the local file system, or connect to any host except the host from which you loaded the applet. The applet security manager enforces these restrictions on applets as protection against applets being able to do unsavory things to unsuspecting users.

If you are trying to use jDriver for Oracle from an applet, then you are violating the first restriction. Your applet will fail when it attempts to load the native (non-Java layer) library that allows jDriver for Oracle to make calls into the non-Java Oracle

client libraries. If you look at the exception that is generated, you will see that your applet fails in `java.lang.System.loadLibrary`, because the security manager determined that you were attempting to load a local library and halted the applet.

You can, however, use the WebLogic JTS or Pool driver for JDBC connectivity in applets. When you use one of these WebLogic multitier JDBC drivers, you need one copy of WebLogic `jDriver` for Oracle (or any other two-tier JDBC driver) for the connection between the WebLogic Server and the DBMS.

Q. Why doesn't my browser applet connect to the database?

Problem: I'm using a WebLogic multitier driver in an applet as an interface to a DBMS. If I run the class using the Sun Appletviewer on my local machine, I have no problems. But when I try to run the applet in a Netscape browser, it will not connect.

A. If Appletviewer works and Netscape does not, it is an indication that you are violating a Netscape security restriction. In this case, the violation is that an applet cannot open a socket to a machine other than the one from which it loaded the applet. To solve this problem, you will have to serve your applet code from the same machine that hosts the DBMS.

In addition, the IP naming format you use in the applet `CODEBASE` and the constructor for the `T3Client` must match. That is, you can't use dot-notation in one place and a domain name in the other.

Q. Why does my applet work with Appletviewer, but not with a browser?

Problem: I tried to run two of the applets in the examples directory of the distribution. I installed the WebLogic classes on my local machine (NT server) and on another machine (a Windows 95 client). I am not using any browsers, just trying to run the applets with Appletviewer. The applets work fine when I run Appletviewer from the NT server, but do not work at all from the Windows 95 client.

A. There are two possible problems: Either the `CODEBASE` tag is not properly set in the applet HTML file, or the class files are not properly loaded on the HTTP server.

The applet works on the NT server because you installed the WebLogic distribution on your NT server. Even if the applet cannot successfully load the necessary classes from the HTTP server, it *does* find them in your local `CLASSPATH`. But when you try to run it from the Windows 95 client, the applet must load the classes over the wire from the HTTP server, and if you haven't installed them correctly, it will fail.

Q. What’s causing ClassFormatErrors with my applet?

Problem: I downloaded your distribution and copied the classes to my HTTP server DocumentRoot. I created an applet that I ran successfully from my Netscape server. I placed it in the server directory `/webz/ns-home/classes/applets/myapp.class` and called it with the following:

```
<APPLET
  CODEBASE=http://myserver.com/webz/ns-home/classes
  CODE=applets.myapp.class>
```

Then I set my attributes in the Administration Console to listen on port 7001, and I started the WebLogic Server on the HTTP machine so I could use my applet with WebLogic JDBC, like this:

```
<APPLET
  CODEBASE=t3://myserver.com:7001/webz/ns-home/classes
  CODE=applets.myapp.class>
```

When I changed the `CODEBASE` tag to point to the WebLogic Server, I started getting ClassFormatErrors.

A. There are several problems with your setup. The most obvious have to do with your `CODEBASE`:

1. The `CODEBASE` tag in your applet should point to your HTTP server, not to WebLogic Server.
2. The directory path referenced in your `CODEBASE` tag is not an absolute directory path on the HTTP server; it is a configured path that originates from the HTTP Document Root. You are using the absolute path in your `CODEBASE` tag. If your class “myapp” is in the “applets” package, then the correct `CODEBASE` for your setup would be:

```
<APPLET
  CODEBASE=http://myserver.com/classes
  CODE=applets.myapp.class>
```

In addition, if you are getting a ClassFormatError, it signals a problem with your HTTP server configuration. It could be that you haven’t loaded the WebLogic or applet classes in the correct directory on the HTTP server, or you are specifying the `CODEBASE` or the `CODE` incorrectly in your `APPLET` tag.

Remember, too, that if you installed the WebLogic distribution on the machine from which you are running the applet, your applet will first look for the WebLogic classes in your local `CLASSPATH`, which may obscure the fact that you haven’t properly

installed the classes for serving from the HTTP server. To test your HTTP configuration properly, you need to temporarily rename the WebLogic classes in your local `CLASSPATH` or try your applet from another machine.

3 FAQs: Clustering

- How do stubs work in a WebLogic Server cluster?
- What happens when a failure occurs and the stub cannot connect to a WebLogic Server instance?
- How does a server know when another server is unavailable?
- How are notifications made when a server is added to a cluster?
- How do clients learn about new WebLogic Server instances?
- How do clients handle DNS requests to failed servers?
- How many WebLogic Servers can I have on a multi-CPU machine?
- Should I use a separate network for multicast in a cluster?
- What should I do if my cluster “hangs” or “freezes”?

Q. How do stubs work in a WebLogic Server cluster?

A. Clients that connect to a WebLogic Server cluster and look up a clustered object obtain a replica-aware stub for the object. This stub contains the list of available server instances that host implementations of the object. The stub also contains the load balancing logic for distributing the load among its host servers.

Q. What happens when a failure occurs and the stub cannot connect to a WebLogic Server instance?

A. When the failure occurs, the stub removes the failed server instance from its list. If there are no servers left in its list, the stub uses DNS again to find a running server and obtain a current list of running instances. Also, the stub periodically refreshes its list of available server instances in the cluster; this allows the stub to take advantage of new servers as they are added to the cluster.

Q. How does a server know when another server is unavailable?

A. WebLogic Server uses two mechanisms to determine if a given server instance is unavailable.

Each WebLogic Server instance in a cluster uses multicast to broadcast regular “heartbeat” messages that advertise its availability. By monitoring heartbeat messages, server instances in a cluster determine when a server instance has failed. The other server instances will drop a server instance from the cluster, if they do not receive three consecutive heartbeats from that server instance

WebLogic Server also monitors socket errors to determine the availability of a server instance. For example, if server instance A has an open socket to server instance B, and the socket unexpectedly closes, server A assumes that server B is offline.

Q. How are notifications made when a server is added to a cluster?

A. The WebLogic Server cluster broadcasts the availability of a new server instance each time a new instance joins the cluster. Cluster-aware stubs also periodically update their list of available server instances.

Q. How do clients learn about new WebLogic Server instances?

A. Once a client has done a JNDI lookup and begins using an object reference, it finds out about new server instances only after the cluster-aware stub has updated its list of available servers.

Q. How do clients handle DNS requests to failed servers?

A. If a server fails and DNS continues to send requests to the unavailable machine, this can waste bandwidth. For a Java client application, this problem occurs only during startup. WebLogic Server caches the DNS entries and removes the unavailable ones, to prevent the client from accessing a failed server twice.

Failed servers can be more of a problem for browser-based clients, because they always use DNS. To avoid unnecessary DNS requests with browser-based clients, use a third-party load-balancer such as Resonate, BigIP, Alteon, and LocalDirector. These products mask multiple DNS addresses as a single address. They also provide more sophisticated load-balancing options than round-robin, and they keep track of failed servers to avoid routing unnecessary requests.

Q. How many WebLogic Servers can I have on a multi-CPU machine?

A. There are many possible configurations and each has its own advantages and disadvantages. BEA WebLogic Server has no built-in limit for the number of server instances that can reside in a cluster. Large, multi-processor servers such as Sun Microsystems, Inc. Sun Enterprise 10000, therefore, can host very large clusters or multiple clusters.

In most cases, WebLogic Server clusters scale best when deployed with one WebLogic Server instance for every two CPUs. However, as with all capacity planning, you should test the actual deployment with your target web applications to determine the optimal number and distribution of server instances. See [Performance Considerations for Multi-CPU Machines](#) for additional information.

Q. Should I use a separate network for multicast in a cluster?

A. No. Multicast traffic is not heavy enough to require a separate network.

Q. What should I do if my cluster “hangs” or “freezes”?

A. If your WebLogic Server cluster “freezes,” you will need to collect certain diagnostic information, including thread dumps and Java garbage collection metrics, before contacting BEA Technical Support. For details, see [Generate a Log File](#).

4 FAQs: Code Examples

- [Where are the examples located?](#)
- [Why won't the examples work?](#)
- [Are the build.cmd and the build.sh scripts still being used?](#)
- [How do I use ANT?](#)

Q. Where are the examples located?

A. Examples, if installed, are located in the `samples\server\src\examples` directory of your WebLogic Server installation and are available from the Start menu.

Q. Why won't the examples work?

A. Each example comes with detailed instructions for building the example class files, configuring the server, and running the example. Make sure that you have completed each of the example instructions.

Usually problems with examples are related to your environment. Here are some troubleshooting hints:

1. If you are using a database, make sure you have run the utility `utils.dbping` to verify that your JDBC driver is correctly installed and configured.
2. Run the `setEnv` script to make sure your `CLASSPATH` is correctly set in the shell or DOS window in which you are running the examples. For more information, see [Setting your development environment](http://www.weblogic.com/docs51/techstart/environment.html) at <http://www.weblogic.com/docs51/techstart/environment.html>.
3. Check the instructions for the examples to make sure you have changed any user-specific variables in the code before compiling.

4. Verify that you are compiling with the `-d` option to direct the class files into the proper directory, as defined in the example instructions.

If the example is an applet, check the `CODE` and `CODEBASE`, and make sure WebLogic Server is running.

For more information, see the WebLogic Server Examples Guide located at `samples/examples/examples.html` in your WebLogic server distribution.

Q. Are the `build.cmd` and the `build.sh` scripts still being used?

A. No. They have been replaced by ANT.

Q. How do I use ANT?

A. Set up the Examples domain environment by running `setExamplesEnv.cmd` in Windows or `setExamplesEnv.sh` in UNIX. Go into your examples directory and type ANT to build the `build.xml` file. If you want to build your own build script, you can pass in the name of the build script by typing `ANT -f myBuild.xml` where `myBuild` is the name of your build script. For more information, see the [Apache website](#).

5 FAQs: Deployment

- How can I set deployment order for applications?
- Can I refresh static components of a deployed application without having to redeploy?
- When should I use the `-nostage` option?
- When should I use the `external_stage` option?
- Can I generate deployment descriptor files automatically?
- Can I set the deployment order for application components? For types of components?
- What is the difference between the `WL_HOME/config/examples/applications` folder and the `WL_HOME/config/examples/stage` folder?
- How do I turn the auto-deployment feature off?
- Where can I learn more about WebLogic 7.0 deployment?

Q. How can I set deployment order for applications?

A. WebLogic Server 7.0 allows you to select the load order for applications. See the ApplicationMBean LoadOrder attribute in [Application](#). WebLogic Server deploys server-level resources (first JDBC and then JMS) before deploying applications. Applications are deployed in this order: connectors, then EJBs, then Web Applications. If the application is an EAR, the individual components are loaded in the order in which they are declared in the `application.xml` deployment descriptor.

Q. Can I refresh static components of a deployed application without having to redeploy?

A. Yes. You can use `weblogic.Deployer` to specify a component and target a server, using the following syntax:

```
java weblogic.Deployer -adminurl http://admin:7001 -name appname
-targets server1,server2 -activate jsp/*.*jsp
```

Q. When should I use the `-nostage` option?

A. Set the staging mode to `-nostage` (using `weblogic.Deployer` or the Administration Console) if you don't want it copied anywhere and want to run it from its present location.

Q. When should I use the `external_stage` option?

A. Set `-external_stage` using `weblogic.Deployer` if you want to stage the application yourself, and prefer to copy it to its target by your own means.

Q. Can I generate deployment descriptor files automatically?

A. Yes, WebLogic Builder automatically generates deployment descriptor files for your J2EE applications. See [WebLogic Builder Online Help](#).

Q. Can I set the deployment order for application components? For types of components?

A. The Deployment Order attribute controls the deployment order of standalone modules and Web Applications relative to other modules and applications of the same type. For example, standalone EJBs with smaller Deployment Order values are deployed before those with higher values.

If your application is an EAR and you are using the two-phase deployment protocol, you can set the application deployment order using the ApplicationMBean LoadOrder attribute.

Q. What is the difference between the `WL_HOME/config/examples/applications` folder and the `WL_HOME/config/examples/stage` folder?

A. The `applications` folder is intended for applications that are not yet ready for a production environment. WebLogic Server dynamically deploys the contents of the `applications` folder. The `stage` folder (or a folder that you create for the same purpose) is for storing copies of deployment files that are ready for deployment in a production environment (deployments that use the `stage` or `external_stage` deployment modes).

Q. How do I turn the auto-deployment feature off?

A. The auto-deployment feature checks the applications folder every three seconds to determine whether there are any new applications or any changes to existing applications and then dynamically deploys these changes.

The auto-deployment feature is enabled for servers that run in development mode. To disable auto-deployment feature, use one of the following methods to place servers in production mode:

- In the Administration Console, click the name of the domain in the left pane, then select the Production Mode checkbox in the right pane.
- At the command line, include the following argument when starting the domain's Administration Server:

```
-Dweblogic.ProductionModeEnabled=true
```

Production mode is set for all WebLogic Server instances in a given domain.

Q. Where can I learn more about WebLogic 7.0 deployment?

A. Here: [Deployment](http://e-docs.bea.com/wls/docs70/deployment.html) at <http://e-docs.bea.com/wls/docs70/deployment.html>.

6 FAQs: EJB

General

- Why is there no polymorphic-type response from a create() or find() method?
- Must EJBs be homogeneously deployed across a cluster? Why?
- What is the free pool?
- Where can I get a copy of the EJB specification?
- Which versions of the EJB specification are supported by the WebLogic Server?

Stateless Session Beans

- How should I set max-beans-in-free-pool for stateless beans?
- How should I set initial-beans-in-free-pool for stateless beans?
- When are stateless EJBs passivated?
- Can I call remove() on a stateless session bean?
- When are ejbCreate and ejbRemove called on stateless EJBs?

Stateful Session Beans

- Can you explain passivation / activation?
- Why did I get a LockTimedOutException?
- What is the difference between the NRU and LRU cache?
- When should I use a Stateful session bean and when should I use a servlet session?
- How big should I make the cache for a stateful session bean?

General Entity Bean Questions

- Can an entity bean be a listener for JMS messages?

CMP Entity Beans

- When are CMP fields loaded? Is it always determined by the finders-load-bean setting? What is the default behavior?
- What's the purpose of the delay-database-insert-until deployment descriptor element?
- Can I map an entity bean to more than one table?
- Can I use a join or intermediate table to implement a One-Many relationship?
- Why can't I hold on to a cmr-field (Container Managed Relationship) collection and use it after the transaction commits?
- Can a foreign key column in the database be mapped to both a cmp-field and a cmr-field?
- Why can't I call the setXXX method for a cmr-field during ejbCreate?
- How can I avoid violating NOT NULL constraints on foreign keys that are mapped to cmr-fields?
- Does WebLogic support auto generating primary keys for entity beans?

Message Driven Beans

- Which security principal does an MDB use to connect to JMS?

EJBs and Transactions

- How should I obtain my JDBC connection so that it participates in the EJB container's transaction?
- My transaction aborted, but my database changes did not rollback.
- Why did my JDBC code throw a rollback SQLException?

Q. Why is there no polymorphic-type response from a create() or find() method?

A. The EJB Specification prohibits this behavior, and the `weblogic.ejb` compiler checks for this behavior and prohibits any polymorphic type of response from a `create()` or `find()` method.

The reason the `create()` and `find()` methods are not polymorphic is similar to the reason constructors are not polymorphic in Java. The derived classes generally do not know or cannot initialize the base class properly.

Q. Must EJBs be homogeneously deployed across a cluster? Why?

A. Yes. Beginning with WebLogic Server version 6.0, EJBs must be homogeneously deployed across a cluster for the following reasons:

- To keep clustering EJBs simple
- To avoid cross server calls which results in more efficiency. If EJBs are not deployed on all servers, cross server calls are much more likely.
- To ensure that every EJB is available locally
- To ensure that all classes are loaded in an undeployable way
- Every server must have access to each EJB's classes so that it can be bound into the local JNDI tree. If only a subset of the servers deploys the bean, the other servers will have to load the bean's classes in their respective system classpaths which makes it impossible to undeploy the beans.

Q. What is the free pool?

A. The free pool is a data structure the EJB container uses to cache anonymous instances of a given bean type. The free pool improves performance by reusing objects and skipping container callbacks when it can.

Q. Where can I get a copy of the EJB specification?

A. You can download it from [Sun's EJB site](http://java.sun.com/products/ejb/index.html). Point your browser to: <http://java.sun.com/products/ejb/index.html>.

Q. Which versions of the EJB specification are supported by the WebLogic Server?

A. The following table summarizes EJB specification support, by WebLogic Server version:

WebLogic Server Version	EJB Version
4.5.x	1.0
5.1	1.1
6.0	1.1 and 2.0 (PFDI)
6.01	1.1 and 2.0
7.0	1.1 and 2.0

Q. How should I set max-beans-in-free-pool for stateless beans?

A. This is explained in our documentation at http://e-docs.bea.com/wls/docs70/////ejb/EJB_environment.html#1135029

Q. How should I set initial-beans-in-free-pool for stateless beans?

A. This is explained in our documentation at http://e-docs.bea.com/wls/docs70/////ejb/EJB_environment.html#1029095

Q. When are stateless EJBs passivated?

A. Stateless ejbs are never passivated. Since stateless ejbs do not have state, there is no need to passivate them. They are put back into the free pool after each method call so they will be available to service other requests.

Q. Can I call remove() on a stateless session bean?

A. Yes. Currently, calling remove() on a stateless session bean is a noop.

Q. When are ejbCreate and ejbRemove called on stateless EJBs?

A. When the stateless beans are created and removed by the EJB container. Note that this does not correspond to calls to create and remove on the stateless home. Depending on your initial-beans-in-free-pool setting, beans may be created by the container during deployment at which point they are placed in the free pool. Aside from during deployment, beans will only be created to service requests when all of the beans in the free pool are in use and the max-beans-in-free-pool limit has not been met. Stateless beans are removed by the EJB container during undeployment.

Q. Can you explain passivation / activation?

A. Passivation and activation are a standard part of a stateful session beans lifecycle. For details, point your browser to:

http://e-docs.bea.com/wls/docs70/////ejb/EJB_environment.html#1028773

Q. Why did I get a LockTimedOutException?

A. When you get a LockTimedOutException while invoking a stateful session EJB, one of two things has occurred:

- You have `<allow-concurrent-calls>` set to true in your `weblogic-ejb-jar.xml` descriptor and your call timed out while waiting to be processed. The timeout used in this case is the value `<trans-timeout-seconds>` element of the `weblogic-ejb-jar.xml` descriptor or its default value of 30 seconds.
- You do not have `allow-concurrent-calls` set to true and you attempt to invoke a stateful session bean that is already busy processing another request. In this case, the second method call will not block and a LockTimedOutException will be thrown immediately.

Q. What is the difference between the NRU and LRU cache?

A. NRU cache works by avoiding passivation as much as possible. Stateful session instances are only passivated when there is memory pressure (your # of beans in the cache approaches the `max-beans-in-cache` size). This is the 'NRU' option in the `weblogic-ejb-jar.xml` and the default behavior. The LRU cache works by passivating a bean after it reached its `idle-timeout-seconds`. So if your `max-beans-in-cache` was 1000 and you only had 10 beans in memory, it would still write the 10 beans to disk after their timeout period expired. This is the 'LRU' option in the `weblogic-ejb-jar.xml`. This was added to 5.1 and 6.x because there were some customers that wrote applications depending on the timeout behavior. This was also the default behavior in 3.1-4.5.

Q. When should I use a Stateful session bean and when should I use a servlet session?

A. The answer to this question is very application-specific and there are situations in which either approach will work. A stateful session bean provides declarative transaction and security checking as well as failover to a secondary in a cluster.

Q. How big should I make the cache for a stateful session bean?

A. The cache for a stateful session bean should usually be equal to the maximum number of concurrent clients of the bean. This is generally a much larger value than

the number of execute threads in the server, therefore, stateful session beans will use more server resources.

Q. Can an entity bean be a listener for JMS messages?

A. No. Message driven beans should be used to consume JMS messages.

Q. When are CMP fields loaded? Is it always determined by the finders-load-bean setting? What is the default behavior?

A. Finders-load-bean is true by default. A bean is retrieved via a finder explicitly if you call a findXXX() method to get the bean and implicitly if you retrieve the bean via a cmr-field getXXX method. Either of these will result in eager loading of the bean's fields if finders-load-bean is true for the bean.

If you aren't calling a finder, but just accessing the bean via a reference you obtained in a different transaction, then fields are always loaded lazily when using 2.0 CMP, i.e. they aren't read from the DBMS during ejbLoad, but rather when a getXXX() method is called. By default there is a single field group that contains all the fields if you don't define any field groups. So, calling a cmp-field getXXX() method loads all of the persistent state of the bean by default.

Q. What's the purpose of the delay-database-insert-until deployment descriptor element?

A. This setting allows you to specify at what point the database insert occurs when creating a CMP bean. Non-null foreign key constraints in your database can make creating CMP beans with relationships tricky. We offer this setting to give developers more flexibility to satisfy those constraints. Also, this setting is necessary to enable the bulk insert feature that was added in WLS 7.0. To enable the bulk insert feature and to provide yourself the maximum amount of flexibility to satisfy any non-null foreign key constraints, we recommend setting the delay-database-insert-until option to commit. For more information on this option, see http://e-docs.bea.com/wls/docs70/////ejb/EJB_environment.html#1118482

Q. Can I map an entity bean to more than one table?

A. In WLS 7.0 it is possible to map an entity bean to multiple tables, however, there are some restrictions. Namely, each table must contain the same primary key columns. The columns can have different names in different tables, but they must contain the same values. When a new bean is created a row will be inserted in each table, and when

a bean is removed a row is removed from each table. The rows in each table are related via their primary key values. The rows mapped to a particular entity bean will always have the same primary key value.

Point your browser to <http://e-docs.bea.com/wls/docs70/////ejb/cmp.html#1093392> for more information.

Q. Can I use a join or intermediate table to implement a One-Many relationship?

A. This is currently not supported. One-Many relationships require that the bean on the Many side of the relationship contain a foreign-key in one of its tables.

Q. Why can't I hold on to a cmr-field (Container Managed Relationship) collection and use it after the transaction commits?

A. This is prohibited by the EJB 2.0 specification. The reason for disallowing this is that the DBMS data backing a cmr-field collection can change in unpredictable ways once your transaction commits. Tracking these changes is difficult and can result in the contents of the cmr-field collection changing when the application doesn't expect it. The bottom line is that developers must retrieve and access cmr-field collections within a single transaction.

Q. Can a foreign key column in the database be mapped to both a cmp-field and a cmr-field?

A. Yes, this has been supported since WLS 6.0 SP1. Note that when the cmp-field is a primary-key field the cmr-field is read-only. In other words, the setXXX method for the cmr-field cannot be used. The value of the primary-key should be initialized as usual in this case. Conversely, when the cmp-field is not a primary-key field, then the cmp-field is read-only. The underlying column is updated via the cmr-field, and the cmp-field just provides a read-only view of the foreign-key.

Q. Why can't I call the setXXX method for a cmr-field during ejbCreate?

A. This is disallowed by the EJB 2.0 specification because the primary-key of the current bean isn't necessarily known during ejbCreate and it may be needed depending on how the relationship is mapped to the underlying DBMS. Cmr-field setters should be called during ejbPostCreate instead of ejbCreate if relationships need to be setup during creation of a bean.

Q. How can I avoid violating NOT NULL constraints on foreign keys that are mapped to cmr-fields?

A. In WLS 7.0, you can set `delay-database-insert-until` to 'commit' and assign the `cmr-field` a value before the current transaction commits. You can also set `delay-database-insert-until` to 'ejbPostCreate' and assign the `cmr-field` a value during `ejbPostCreate`.

Q. Does WebLogic support auto generating primary keys for entity beans?

A. Yes, this feature was added in WLS 6.1. For more information, see the [DTD comments for the <automatic-key-generation> element](#). Point your browser to:

<http://www.bea.com/servers/wls700/dtd/weblogic-ejb-jar.dtd>.

Q. Which security principal does an MDB use to connect to JMS?

A. As of WLS 6.1 SP2, an MDB uses the same principle to connect to JMS as it does to process messages. This is either the principle that is mapped to the run-as role specified for the bean, or 'guest' if no run-as role is provided. Prior to WLS 6.1 SP2, this behavior was not well defined.

Q. How should I obtain my JDBC connection so that it participates in the EJB container's transaction?

A. You must get your JDBC connection from a `TxDataSource` or from the JTS driver. If you get a JDBC connections from a plain `DataSource` or directly from the JDBC driver, it will NOT participate in the EJB container transaction.

`TxDataSources` are recommended instead of accessing the JTS driver directly. `TxDataSources` are the standard mechanism in JDBC 2.0 for connection pooling, and they support 2PC/XA transactions.

Q. My transaction aborted, but my database changes did not rollback.

A. See previous question. You must obtain your JDBC connections from a `TxDataSource`.

Q. Why did my JDBC code throw a rollback `SQLException`?

A. Your JDBC code may throw the following exception:

```
"The coordinator has rolled back the transaction.  
No further JDBC access is allowed within this transaction."
```

The WebLogic JTS JDBC driver throws this exception when the current JDBC connection transaction rolls back prior to or during the JDBC call. This exception indicates that the transaction in which the JDBC connection was participating was rolled back at some point prior to or during the JDBC call.

The rollback may have happened in an earlier EJB invoke that was part of the transaction, or the rollback may have occurred because the transaction timed out. In either case, the transaction will be rolled back, the connection returned to the pool and the database resources released. In order to proceed, the JTS JDBC connection must be closed and reopened in a new transaction.

7 FAQs: Installation

- Which platforms are available for WebLogic Server?
- I downloaded the WebLogic Server installation file, but the installation program will not run. What should I do?
- Do I need to install WebLogic Server as root on Solaris?
- Can I run the Configuration Wizard outside the installer?
- I love the new domain configuration wizard. Can I create my own templates?

Q. Which platforms are available for WebLogic Server?

A. See the [BEA WebLogic Server Platform Support](http://e-docs.bea.com/platform/suppconfigs/index.html) page at <http://e-docs.bea.com/platform/suppconfigs/index.html>.

Q. I downloaded the WebLogic Server installation file, but the installation program will not run. What should I do?

A. The installation file may have been corrupted during the download. Run a checksum on the installation file and check with technical support for the proper values.

Q. Do I need to install WebLogic Server as root on Solaris?

A. No you don't need to be root, depending on directory permissions.

Q. Can I run the Configuration Wizard outside the installer?

A. Yes. You can start the Configuration Wizard from the Start menu or using a script in the `utils` directory. See [Creating New Domains Using the Configuration Wizard](#).

Q. I love the new domain configuration wizard. Can I create my own templates?

A. Support for custom templates for the Configuration Wizard is coming soon. Currently, it is an undocumented, unsupported feature.

8 FAQs: Java

- Can you help me debug my program?
- Where can I get help learning Java?
- Where do I get the JDK?
- How do I set up my CLASSPATH?
- Why won't the examples work?
- Where can I get help with Java error messages?
- Why did a client-server message generate a StackOverflowException?
- Will a JIT make my Java application run faster?
- Can I redistribute the JDK that is bundled with WebLogic Server?

Q. Can you help me debug my program?

A. If the problem you are having is not directly related to our software, we suggest that you use a Java development tool that helps you with debugging, and that you invest in some books or training to help you learn Java. There are many ways to build debugging into your program, and getting good training in Java programming is a good start to understanding how to do this.

Q. Where can I get help learning Java?

A. There are a lot of books and online references to Java. A good starting place is at the JavaSoft website [documentation index](http://www.javasoft.com/docs/index.html) at <http://www.javasoft.com/docs/index.html>, which has links to white papers and the Java Tutorial. You can find books on Java at any of the major online book shopping sites.

Q. Where do I get the JDK?

A. WebLogic 6.1 bundles JDK131. See our Platform support page for information about specific JDKs that we have tested and certified for use with WebLogic software.

Once you determine which version of the JDK you will be using, go to the [JavaSoft](http://www.javasoft.com/products/) website at <http://www.javasoft.com/products/>. Many platform vendors provide an optimized version of the JDK for their computers.

Q. How do I set up my CLASSPATH?

A. Setting up your CLASSPATH correctly depends on what you are trying to do. The most common tasks are described below:

- **Starting WebLogic Server.** See Setting the Classpath Option in the [Starting and Stopping WebLogic Servers](#) section of the *Administration Guide* at <http://e-docs.bea.com/wls/docs70/adminguide/startstop.html>. In addition, your WebLogic distribution includes shell scripts that you can use to start the server. These scripts, which are located in the domain directories under the `config` directory of your WebLogic Server distribution, automatically set up the CLASSPATH variable in the shell before starting the server.
- **Compiling Application Classes or Using WebLogic Server Utilities.** See Setting the Classpath for Compiling in the [Developing WebLogic Server Components](#) section of *Developing WebLogic Server Applications*.
- **Working With WebLogic Server Code Examples.** See the WebLogic Server Examples Guide located at `samples/examples/examples.html` in your WebLogic Server distribution.

Q. Why won't the examples work?

A. Usually problems with examples are related to your environment. Here are some troubleshooting hints:

1. If you are using a database, make sure you have run the utility `utils.dbping` to verify that your JDBC driver is correctly installed and configured.
2. Run the `setEnv` script to make sure your CLASSPATH is correctly set in the shell or DOS window in which you are running the examples. For more information, see [Setting your development environment](#) at <http://www.weblogic.com/docs51/techstart/environment.html>.

3. Check the instructions for the examples to make sure you have changed any user-specific variables in the code before compiling.
4. Verify that you are compiling with the `-d` option to direct the class files into the proper directory, as defined in the example instructions.

If the example is an applet, check the CODE and CODEBASE, and make sure WebLogic Server is running.

Q. Where can I get help with Java error messages?

A. Many questions we receive at BEA are related to generic Java error messages and are not specific to WebLogic Server. Here are some links that contain helpful information about Java error messages.

Table 8-1

Source	Description
Sun's Java Developer Connection at <code>http://developer.java.sun.com/developer/</code>	This forum includes Questions and Answers on a wide variety of Java topics, including error messages. Use the Search box for fast results. For example, type "classpath error" in the Search box.
Sun's Java APIs at <code>http://java.sun.com/products/</code>	Check the Java API to see if there is an exception description for the class you are using.

Q. Why did a client-server message generate a StackOverflowException?

A. If you are sending a particularly large data structure using `java.io.Serialization`, you may exceed the per-thread size limit for either the Java or native stack. You can increase the stack size by using the following command line options:

- `-ss Stacksize` to increase the native stack size or
- `-oss Stacksize` to increase the Java stack size,

where *Stacksize* is expressed as an integer followed by "k" or "m" for kbytes or mbytes. For example,

```
$java -ss156k (native)
$java -oss600k (Java)
```

The default *native* stack size is 128k, with a minimum value of 1000 bytes. The default *java* stack size is 400k, with a minimum value of 1000 bytes.

Q. Will a JIT make my Java application run faster?

A. A Just-In-Time compiler will make some Java applications run faster. A JIT works by storing generated machine code in memory and reusing it when possible. For example, if you execute the same operation 1000 times in a loop, a JIT will improve performance of this operation since the code will only be generated once. Applications with a lot of native methods will not see as much performance improvement as pure-Java applications.

If you use a JIT, you may want to turn off the JIT during debugging to facilitate stacktracing. If you are doing performance testing with a JIT, make sure that you execute the same test several times in the same invocation and then throw away the first result to get an idea of how long the transaction will take when your application is running in a steady state. The first time the code is executed, your test will take longer (the “code generation hit”).

Q. Can I redistribute the JDK that is bundled with WebLogic Server?

A. BEA Systems has the non-exclusive right to grant a third party, such as an independent software vendor (ISV), the right to redistribute the JDK that is bundled with WebLogic Server without any modifications of any kind. The following are caveats to this general statement:

- The ISV cannot remove or alter any proprietary legends or notices contained in the JDK. The ISV shall not decompile, disassemble, decrypt, extract, or otherwise reverse engineer or modify the JDK. The JDK may not be leased or assigned in whole or in part.
- The ISV must enter into a signed agreement with its distributors on terms substantially similar to those contained here in this redistribution policy explanation.
- The ISV requires an end user license agreement with the product within which it embeds WebLogic Server.
- The embedding of the JDK in ISV products does not include maintenance and support of the JDK by the JDK Provider. BEA Systems shall be solely responsible for providing maintenance and support for its ISVs and distributors. The ISV shall be solely responsible for providing maintenance and support for the end users of its products.

-
- If an ISV wants to ship a JDK that is different from the JDK that BEA ships with WebLogic Server, that ISV needs to get those bundling rights directly from Sun or HP. For example:

Assume that BEA ships WebLogic Server 6.0 with JDK 1.3 only, that BEA ships WebLogic Server 5.1 with JDK 1.1 only, and that an ISV wants to ship JDK 1.1 with WebLogic Server 6.0 in an integrated product offered by that ISV. Unless BEA, for BEA business reasons, elects to make WebLogic Server 6.0 with JDK 1.1 generally available, the ISV couldn't ship JDK 1.1 with WebLogic Server 6.0 in an integrated product offered by that ISV under BEA's agreement with that ISV or under BEA's agreement with Sun. The ISV could, however, obtain its own binary distribution agreement for the JDK from Sun and under that agreement bundle JDK 1.1 with its value added software solution consisting of the ISV applications and WebLogic Server 6.0 integrated.

9 FAQs: J2EE Connector Architecture

- Why do I get the following exception when viewing the JNDI tree?
- Is it possible to use the Oracle database rather than PointBase with the current implementation of the WebLogic J2EE Connector Architecture?
- When deploying a resource adapter (.rar) to WebLogic Server, are its classes placed in the WebLogic classpath?
- Why does WebLogic Server invoke the `ManagedConnection.addConnectionEventListener()` function whenever the sample EJB calls `ConnectionFactory.getConnection()` to connect to the EIS?
- Why do I get an exception when I compile my EJB to use a Resource Adapter that supports CCI?
- The `BEA.com.bea.adapter.dbms.cci.ConnectionImpl` does not directly implement `javax.resource.cci.Connection`. Is there a work-around for this?

Q. Why do I get the following exception when viewing the JNDI tree?

```
isSerializable(class.javax.naming.Binding)
java.io.NotSerializableException:
java.io.PrintWriter at
java.io.ObjectOutputStream.writeObject
```

A. The Weblogic Server JNDI implementation requires objects to be serializable, not referencable. A `PrintWriter` cannot be serialized and therefore should be declared `transient`.

Q. Is it possible to use the Oracle database rather than PointBase with the current implementation of the WebLogic J2EE Connector Architecture?

A. The sample provided contains a resource adapter that represents any database system. By default, configurations are set to use PointBase. In particular, the configuration properties in the `weblogic-ra.xml` file set a PointBase datasource. This can be replaced with Oracle settings.

Also, the resource adapter (in particular the `ManagedConnectionFactory`) must be implemented to support Oracle. The resource adapter provided in this sample makes use of the JDBC and therefore can support any database system which the resource adapter is configured to represent.

Q. When deploying a resource adapter (.rar) to WebLogic Server, are its classes placed in the WebLogic classpath?

For instance, I am deploying an EJB and a resource adapter (.rar), the EJB has no dependencies on the .rar because the EJB is writing to the common client interface (CCI). The EJB client application has sends/marshals as parameter classes that are defined in the .rar. For some reason the EJB's class loader hierarchy cannot find the definition of this .rar-specific class, even though the .rar is deploying successfully. I receive the following error on the EJB client:

```
java.rmi.UnmarshalException: error unmarshalling arguments; nested exception
is:
java.lang.ClassNotFoundException:
com.mycompany.InteractionSpecImpl
```

A. When you pass an instance of `com.myclientcompany.server.eai.InteractionSpecImpl` as an argument to your EJB, the `appServer` needs to de-serialize (unmarshal) the object under the EJB context, and it needs the required class for unmarshalling, inside the `ejb-jar(raTester.jar)`. So if you include the `interactionspecimpl` class in your `ejb-jar` file, then you do not need to include those classes in your server's classpath.

Q. Why does WebLogic Server invoke the `ManagedConnection.addConnectionEventListener()` function whenever the sample EJB calls `ConnectionFactory.getConnection()` to connect to the EIS?

A. This is a requirement and is part of the contract between the Resource Adapter and the application server.

Q. Why do I get an exception when I compile my EJB to use a Resource Adapter that supports CCI?

I specify `javax.resource.cci.ConnectionFactory` in the `resource-ref` but I get the following exception when I attempt to compile my EJB:

```
weblogic.xml.process.SAXValidationException:  
ejb-jar.enterprise-beans.session.resource-ref.res-type.  
must be one of the values:  
javax.sql.DataSource, javax.jms.QueueConnectionFactory,  
javax.jms.TopicConnectionFactory,  
java.net.URL,  
javax.mail.Session  
at  
weblogic.ejb20.dd.xml.EjbJarLoader_EJB11.__post_84
```

A. Make sure your `ejb-jar.xml` is referencing the EJB2.0 DTD instead of the EJB1.1 DTD. The `ConnectionFactory` resource ref is only supported in the EJB 2.0 DTD.

Q. The BEA `com.bea.adapter.dbms.cci.ConnectionImpl` does not directly implement `javax.resource.cci.Connection`. Is there a work-around for this?

A. Yes. The BEA `com.bea.adapter.dbms.cci.ConnectionImpl` extends `com.bea.adapter.cci.AbstractConnection`; this in turn implements the `Connection` interface. The Proxy is constructed using the interfaces from the most-derived class (`ConnectionImpl`). The `dumpFamilyTree` printout shows that the `getInterfaces` call on the `ConnectionImpl` class does NOT include the `Connection` interface. However, the `getInterfaces` call on `AbstractConnection` does include the `Connection` interface.

The work-around is that the `ConnectionImpl` class must directly implement the interface class specified in the `ra.xml` file. (This might be a redundant statement in the code, particularly if it extends a class that has already implemented the class.) Then rebuild the adapter and try your test again.

10 FAQs: WebLogic JDBC

- When should I use a TxDataSource instead of a DataSource?
- Why do I get "java.sql.SQLException: weblogic.common.ResourceException: Access not allowed" when trying to connect to a DataSource in WebLogic Server 7.0?
- Can I enable requests to a JDBC connection pool for a database connection to wait until a connection is available?
- How can I avoid ResourceExceptions when sending more requests for database connections from the pool than are currently available?
- How should I use MultiPools?
- How can I tell if a database is unavailable?
- Can the WebLogic JDriver for Microsoft SQL Server connect to the database server using a trusted connection on NT/WIN2K?
- Can I use the PointBase DBMS included with WebLogic Server for development or production?

Q. When should I use a TxDataSource instead of a DataSource?

A. If your applications or environment meet any of the following criteria, you should use a Tx Data Source instead of a Data Source:

- Uses the Java Transaction API (JTA)
- Uses the EJB container in WebLogic Server to manage transactions
- Includes multiple database updates within a single transaction
- Accesses multiple resources, such as a database and the Java Messaging Service (JMS), during a transaction

- Uses the same connection pool on multiple servers

With an EJB architecture, it is common for multiple EJBs that are doing database work to be invoked as part of a single transaction. Without XA, the only way for this to work is if all transaction participants use the exact same database connection. WebLogic Server uses the JTS driver and a TxDataSource (with Emulate Two-Phase Commit for non-XA Driver selected) to do this behind the scenes without requiring you to explicitly pass the JDBC connection from EJB to EJB. With XA (requires an XA driver), you can use a Tx Data Source in WebLogic Server for distributed transactions with two-phase commit so that EJBs can use a different database connections for each part of the transaction. In either case (with or without XA), you should use a Tx Data Source.

Q. Why do I get "java.sql.SQLException: weblogic.common.ResourceException: Access not allowed" when trying to connect to a DataSource in WebLogic Server 7.0?

A. You need to create the correct ACL for the DataSource following this procedure in the Administration Console:

1. Compatibility Security => ACLs

Create a new ACL:

name : weblogic.jdbc.connectionPool.yourPoolname

permission : admin

group : Administrators

2. Compatibility => click Refresh button

3. Services => JDBC => Connection Pool

Create a new Connection Pool:

ACL Name : weblogic.jdbc.connectionPool.yourPoolname

In 'Target' tab, choose server and click the Apply button.

4. Services => JDBC => Data Sources

You can create a new Data Source using this connection pool successfully

Q. Can I enable requests to a JDBC connection pool for a database connection to wait until a connection is available?

A. No, there's no way to allow a request to wait for a pool connection, and from the system point of view there should not be. Each requests that waits for a connection ties up one of the fixed number of execute threads in the server, which could otherwise be

running another server task. Too many waiting requests could tie up all of the execute threads and freeze the server.

Q. How can I avoid ResourceExceptions when sending more requests for database connections from the pool than are currently available?

A. The fundamental problem is too few resources (database connections in the connection pool) for the work load. The correct response is to increase the maximum number of connections in the connection pool. Optimally designed applications only require the server to have one pool connection per execute thread.

The proper application response to a resource exception is *not* to retry the request in a tight loop, which would tie up execute threads on the server.

You should design your application to gracefully fail if no connections are available. Try to ensure that you get the connection as late as possible in your application code and return them to the pool as early as possible so that you do not see as many NoResource exceptions. It is better to have the connection as a method level variable and close the connection in a finally block as in the following example:

```
try{
    ...
} catch(Exception handleEx) {
    ...
} finally {
    try{ conn.close();
} catch (Exception ignore){} // always return the connection to pool
}
```

Q. How should I use MultiPools?

A. You can use MultiPools in one of two ways 1) for high availability in the event a database connection fails, or 2) for load balancing MultiPool. Because you can choose only one option, you need to determine the primary purpose of your MultiPool. For more information, see Using Multipools in the [Configuring and Administering WebLogic JDBC Features](#) section of *Programming WebLogic JDBC* at <http://e-docs.bea.com/wls/docs70/jdbc/programming.html>.

Q. How can I tell if a database is unavailable?

A. Fundamentally, there is no way to tell if a database has gone down except by trying to make a connection and failing.

Furthermore, a database can become unavailable at any time after you make and use a connection. We recommend that you write your code to be able to handle unexpected failures, which can come in any form depending on what the client is doing when the database goes down.

WebLogic Server does provide the `dbping` utility to test the connection between WebLogic Server and your DBMS using a JDBC driver. See `dbping` in “Using the WebLogic Java Utilities” in the *Administration Guide*.

Q. Can the WebLogic JDriver for Microsoft SQL Server connect to the database server using a trusted connection on NT/WIN2K?

A. No, the WebLogic JDriver for Microsoft SQL Server does not support trusted connections. Note that the WebLogic JDriver for Microsoft SQL Server is deprecated and will be removed in WebLogic Server 8.1.

Q. Can I use the PointBase DBMS included with WebLogic Server for development or production?

A. PointBase Server is an all-Java DBMS product included in the WebLogic Server distribution solely in support of WebLogic Server evaluation, either in the form of custom trial applications or through packaged sample applications provided with WebLogic Server. Non-evaluation development and/or production use of the PointBase Server requires a separate license be obtained by the end user directly from PointBase.

11 FAQs: WebLogic jDriver for MSSQL Server

- Can the Weblogic JDriver for MSSQL Server connect to the database server using a trusted connection on NT/WIN2K?
- How do I connect to an SQL Server instance that is running on a machine with multiple instances of SQL Server 2000?

Q. Can the Weblogic JDriver for MSSQL Server connect to the database server using a trusted connection on NT/WIN2K?

A. Our driver doesn't support trusted connections.

Q. How do I connect to an SQL Server instance that is running on a machine with multiple instances of SQL Server 2000?

A. Each instance of MS SQL Server must be listening on a different port. So, you can use the port number in the properties that you pass to the `getConnection()` method or, in case of connection pools, you can specify the port property in the following properties:

```
server=machineName  
port=instancePort
```

To find the port number where each MS SQL Server instance is running, run the server network utility (in the Microsoft SQL Server program group), select the server instance, select TCP/IP, and click the properties button.

12 FAQs: WebLogic jDriver for Oracle

- Why does FOR UPDATE in Oracle 8 cause an ORA-01002 error?
- What causes an OCIW32.dll error?
- What transaction isolation levels does the WebLogic jDriver for Oracle support?
- How do I use Unicode codesets with the WebLogic jDriver for Oracle driver?
- How do I use OS Authentication with WebLogic jDriver for Oracle and Connection Pools?
- What type of object is returned by ResultSet.getObject()?
- How do I limit the number of Oracle database connections generated by WebLogic Server?
- How do I call Oracle stored procedures that take no parameters?
- How do I bind string values in a PreparedStatement?
- Why do I get unexpected characters from 8-bit character sets in WebLogic jDriver for Oracle?
- How do I learn what codesets are available in Oracle?
- How do I look up an "ORA" SQLException?
- What is error "ORA-6502?"
- Why do I get an error while trying to retrieve the text for ORA-12705?
- Why do I run out of resources during updates with Oracle's database link?

- Why do I get error "ORA-03120" when trying to access multi-byte characters from the CLOB/NCLOB column?
- Why does executing the PreparedStatement class cause a "TRUNC fails: ORA-00932: inconsistent datatypes" error?
- Why do I get an "ORA-01400: Cannot insert NULL into column name" when inserting a blank string?

Q. Why does FOR UPDATE in Oracle 8 cause an ORA-01002 error?

A. The Oracle 8 server generates an **ORA-01002: fetch out of sequence** error message when you use a FOR UPDATE statement with AUTOCOMMIT turned **on** (which is the default state when using JDBC). This is known to happen on Oracle 8.0 and 8.1 on Solaris and on Oracle 8.1 on Windows NT. If you turn AUTOCOMMIT **off**, you will not receive this error. Because this problem is due to a change in the Oracle 8 server, you should contact Oracle support for more information.

Q. What causes an OCIW32.dll error?

A. You may receive the following error message when using your JDBC driver for Oracle: "The ordinal 40 could not be loaded in the dynamic link library OCIW32.dll." This problem is caused by an out-of-date version of OCIW32.DLL in your system directory. Some programs install this file in the system directory in order to run. If you remove this file from the system directory you should no longer receive this error.

Q. What transaction isolation levels does the WebLogic jDriver for Oracle support?

A. Your servlet application may use Oracle Thin Drivers to access a database that includes BLOB fields. If you install and try to use WebLogic jDriver for Oracle and the same code fails and produces an exception similar to the following:

```
com.roguewave.jdbtools.v2_0.LoginFailureException:
TRANSACTION_READ_UNCOMMITTED isolation level not allowed
  The Stack Trace:
com.roguewave.jdbtools.v2_0.LoginFailureException:
TRANSACTION_READ_UNCOMMITTED isolation level not allowed
  at
  com.roguewave.jdbtools.v2_0.jdbc.JDBCServer.createConnection
(JDBCServer.java :46)
  at com.roguewave.jdbtools.v2_0.ConnectionPool.getConnection_
(ConnectionPool.java :412)
  at com.roguewave.jdbtools.v2_0.ConnectionPool.getConnection
(ConnectionPool.java :109)
```

Setting the `Isolation_level` to `1` in the code that calls the `RogueWave JDBCServer` class works with the Oracle thin driver but fails with `WebLogic jDriver` for Oracle.

`WebLogic jDriver` for Oracle supports the following transaction isolation levels:

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
```

According to the Oracle documentation, the Oracle DBMS only supports these two isolation levels. Unlike other JDBC drivers, `WebLogic's` drivers throw an exception if you try to use an isolation level that is unsupported. Some drivers silently ignore attempts to set an unsupported isolation level. `WebLogic` suggests testing whether the Oracle thin driver is not just ignoring settings for unsupported isolation events.

Q. How do I use Unicode codesets with the `WebLogic jDriver` for Oracle driver?

A. To use Unicode codesets:

1. Install the appropriate codeset when you install Oracle. If you did not do this in the original installation, you will need to re-run the Oracle installer and install the proper codeset.
2. Define the `NLS_LANG` variable in the environment where the JDBC driver is running. Do this by assigning the proper codeset to `NLS_LANG` in the shell from where you start the `WebLogic Server`.

The `Developers Guide` has more information about internationalization support. For general information about Unicode see the [Unicode Web site](http://www.unicode.org) at <http://www.unicode.org>. For a list of Unicode language abbreviations, see the [JavaSoft Web site](http://java.sun.com/products/jdk/1.2/docs/guide/internat/encoding.doc.html) at <http://java.sun.com/products/jdk/1.2/docs/guide/internat/encoding.doc.html>.

Q. How do I use OS Authentication with `WebLogic jDriver` for Oracle and Connection Pools?

A. Using OS authentication in connection pools essentially means that you are using the `UserId` of the user who started `WebLogic Server`. OS authentication is available on Windows and UNIX. This means that database security will rely strictly on the security of `WebLogic`; that is, if you are allowed to make a client connection to the `WebLogic Server` and access the pool, then you can get to the database.

You can do this with `WebLogic jDriver` for Oracle because Oracle uses the process owner to determine who is attempting the connection. In the case of `WebLogic JDBC`, this is always the user that started the `WebLogic Server`.

12 FAQs: WebLogic jDriver for Oracle

To set up your Oracle instance to use this feature, your DBA needs to follow these basic steps. The full procedure is described in more detail in your Oracle documentation.

1. Add the following line to the INIT[sid].ORA file:

```
OS_AUTHENT_PREFIX = OPSS
```

Note that the string “OPSS” is arbitrary and up to the DBA.

2. Log in to the Oracle server as SYSTEM.
3. Create a user named `OPSS$userid`, where *userid* is some operating system login ID. This user should be granted the standard privileges (for example, CONNECT and RESOURCE).
4. Once the *userid* is set up, you can connect with WebLogic jDriver for Oracle by specifying “/” as the username property and “” as the password property. Here is an example for testing this connection with the `dbping` utility:

```
$ java utils.dbping ORACLE "/" "" myserver
```

Here is a code example for WebLogic jDriver for Oracle:

```
Properties props = new Properties();
props.put("user",    "/");
props.put("password", "");
props.put("server",  "myserver");

Class.forName("weblogic.jdbc.oci.Driver").newInstance();
Connection conn = myDriver.connect("jdbc:weblogic:oracle",
    props);
```

5. Use the Administration Console to set the attribute for your connection pool. The following code is an example of a JDBC connection pool configuration using the WebLogic jDriver for Oracle:

```
<JDBCConnectionPool
  Name="myPool"
  Targets="myserver,server1"
  DriverName="weblogic.jdbc.oci.Driver"
  InitialCapacity="1"

  MaxCapacity="10"
  CapacityIncrement="2"
  Properties="databaseName=myOracleDB"
```

Q. What type of object is returned by `ResultSet.getObject()`?

A. WebLogic JDBC driver for Oracle always returns a Java object that preserves the precision of the data retrieved. WebLogic JDBC driver for Oracle returns the following from the `getObject()` method:

- For columns of types `NUMBER(n)` and `NUMBER(m,n)`: a `Double` is returned if the defined precision of the column can be represented by a `Double`; otherwise `BigDecimal` is returned.
- For columns of type `NUMBER`: Because there is no explicit precision, the Java type to return is determined based on the actual value in each row, and this may vary from row to row. An `Integer` is returned if the value has a zero-valued fractional component and the value can be represented by an integer.

For example, 1.0000 will be an integer. A `long` is returned for a value such as 123456789123.00000. If a value has a non-zero fractional component, a `Double` is returned if the precision of the value can be represented by a `Double`; otherwise a `BigDecimal` is returned.

Q. How do I limit the number of Oracle database connections generated by WebLogic Server?

A. You can use connection pools to limit the number of Oracle database connections generated by WebLogic Server in response to client requests. Connection pools allow T3 applications to share a fixed number of database connections. For information on how to set up connection pools, see [Programming WebLogic JDBC](http://e-docs.bea.com/wls/docs70/jdbc.html) at <http://e-docs.bea.com/wls/docs70/jdbc.html>.

Q. How do I call Oracle stored procedures that take no parameters?

A. Here is what we use that works:

```
CallableStatement cstmt = conn.prepareCall("Begin procName;  
      END;");  
cstmt.execute();
```

where *procName* is the name of an Oracle stored procedure. This is standard Oracle SQL syntax that works with any Oracle DBMS. You might also use the following syntax:

```
CallableStatement cstmt = conn.prepareCall("{call procName;}");  
cstmt.execute();
```

This code, which conforms to the Java Extended SQL spec, will work with any DBMS, not just Oracle.

Q. How do I bind string values in a PreparedStatement?

A. Suppose you are trying to get the PreparedStatement class to bind Strings in a statement. The `setString()` method doesn't seem to work. Here is how you have set up the PreparedStatement:

```
String pstmt = "select n_name from n_table where n_name LIKE '??'";
PreparedStatement ps = conn.prepareStatement(pstmt);
ps.setString(1, "SMIT");
ResultSet rs = ps.executeQuery();
```

The preceding code does not work because the complete value needs to be specified in a String (without using embedded quotes) and then bound to an unquoted question-mark (?). Here is the corrected code:

```
String matchvalue = "smit%";
String pstmt = "select n_name from n_table where n_name LIKE ?";
PreparedStatement ps = conn.prepareStatement(pstmt);

ps.setString(1, matchvalue);
ResultSet rs = ps.executeQuery();
```

Q. Why do I get unexpected characters from 8-bit character sets in WebLogic jDriver for Oracle?

A. If you are using an Oracle database with an 8-bit character set on Solaris, make sure you set `NLS_LANG` to the proper value on the client. If `NLS_LANG` is unset, it defaults to a 7-bit ASCII character set, and tries to map characters greater than ASCII 128 to a reasonable approximation (for example, `´`, ```, `â` would all map to a). Other characters are mapped to a question mark (?).

Q. How do I learn what codesets are available in Oracle?

A. To find out what codesets you currently have available in Oracle, execute the following SQL query from SQLPlus at the command line:

```
SQL> SELECT value FROM v$nls_valid_values
        WHERE parameter='CHARACTERSET';
```

The response lists of all codesets currently installed on your system. This listing will look something like the following shortened list:

VALUE

US7ASCII
WE8DEC
WE8HP
US8PC437
WE8EBCDIC37
WE8EBCDIC500
WE8EBCDIC285
...

If you want to constrain the value in the query to a specific codeset you are searching for, you might use a SQL query like the following:

```
SQL> SELECT value FROM v$nls_valid_values  
       WHERE parameter='CHARACTERSET' and VALUE='AL24UTFSS';
```

This would produce the following response if the codeset is installed:

VALUE

AL24UTFSS

You can use Oracle's installation tools to install additional codesets. Contact [Oracle](http://www.oracle.com/) at <http://www.oracle.com/> for more information.

Q. How do I look up an "ORA" SQLException?

A. If your WebLogic jDriver for Oracle application produces an SQLException such as:

```
java.sql.SQLException: ORA-12536: TNS: operation would block
```

You can look up an Oracle error by using the `oerr` command. For example, the description of error ORA-12536 can be found with the command:

```
> oerr ora 12536
```

Q. What is error "ORA-6502?"

A. The default length of a string bound to an `OUTPUT` parameter of a `CallableStatement` is 128 characters. If the value you assign to the bound parameter exceeds that length, you will get this error.

You can adjust the length of the value of the bound parameter by passing an explicit length with the `scale` argument to the

`CallableStatement.registerOutputParameter()` method.

Q. Why do I get an error while trying to retrieve the text for ORA-12705?

A. This error occurs when you have not set the `ORACLE_HOME` environment variable properly — a common mistake. In order to use WebLogic jDriver for Oracle, the Oracle client software needs to be installed and `ORACLE_HOME` must be set.

You may also see this error message if you try to use WebLogic jDriver for Oracle's internationalization capabilities with a language/codeset combination that is not installed on your system. If you get the ORA-12705 error with the correct error text, then either you have set `NLS_LANG` improperly, or you do not have the right codesets installed on your system.

Q. Why do I run out of resources during updates with Oracle's database link?

A. When you use Oracle's database link to update your database, you may get error "maximum number of temporary table locks exceeded" even if you close your result sets and statements when you finish.

The database link is an object in the local database that allows you to access tables, views, and such in a remote database. The database link is controlled by the Oracle server, so the driver has no control over its use of resources. The link appears to perform the commit (since other processes could see the records that were being created), but it doesn't free any resources until the connection is closed. The solution is to remove the database link and use the JDBC driver to do your selects, inserts, and updates.

Q. Why do I get error "ORA-03120" when trying to access multi-byte characters from the CLOB/NCLOB column?

A. Accessing exceeded characters would result in ORA-03120 error when getting the length of the clob from the CLOB/NCLOB column returns a bigger value than the actual length from the OCI layer. Using Oracle 8.1.6.3 solves this problem.

Q. Why does executing the PreparedStatement class cause a "TRUNC fails: ORA-00932: inconsistent datatypes" error?

A. According to Oracle Metalink Bug Database Doc ID: 144784.1, in the absence of explicit data typecasting, OCI assumes that a bind variable is a CHAR data type. If the SQL statement intends to use the bind variable as a DATE data type, but OCI thought

it was a CHAR, the SQL parser will have a conflict in data types. The fix is to explicitly use data conversion functions to convert the bind variables in the problem queries. For example, a select string of

```
String st = "select count(*) from simple_table where  
TRUNC(mydate) = TRUNC(?)";
```

should be changed to:

```
String st = "select count(*) from simple_table where  
TRUNC(mydate) = TRUNC(TO_DATE(?))";
```

Q. Why do I get an “ORA-01400: Cannot insert NULL into *column name*” when inserting a blank string?

A. This is a known Oracle issue. When inserting or updating a value for a varchar2, if you try to insert an empty string (“”), Oracle interprets the value as NULL. If there is a NOT NULL restriction on the column in which you are inserting the value, the database throws the ORA-01400 error.

13 FAQs: JMS

The WebLogic JMS Product

- What makes WebLogic JMS unique?
- Which JMS features are new in WebLogic Server 7.0?
- Where can I learn more about WebLogic JMS?
- Are there C/C++ interfaces to WebLogic JMS?
- Is there a smaller version of the `weblogic.jar` file for supporting clients?

Configuration

- How do I start WebLogic Server and configure JMS?
- How do I configure WebLogic JMS security?
- Can I still use the default connection factories supported in WebLogic JMS 5.1?
- Why does `JMSSession.createTopic` or `JMSSession.createQueue` fail to create a destination in WebLogic JMS 7.0? (It worked in version 5.1?)
- How do I programmatically get a list of Queues or Topics?
- How do I use a temporary destination?
- How do I use MBeans to print runtime statistics?
- Can two JMS servers share the same persistent store?

Persistent Stores

- Which types of JDBC databases does WebLogic JMS support?
- How do I use a third-party JDBC driver with WebLogic JMS?

- What if my JDBC database becomes corrupt?
- How do I use persistence?
- How does a file store compare with a JDBC store?

Administration

- How do the JMS server/destination message maximum and threshold values work?
- How do I configure JDBC so that the JMS JDBC Store recovers automatically?
- What is the value of WebLogic JMS clustering in version 7.0?
- How can I control on which WebLogic Server(s) my application will run?
- How do I perform a manual fail-over?
- Does the WebLogic JMS server find out about closed or lost connections, crashes, and other problems and does it recover from them?
- Do I need to use the WLS T3 protocol?
- How do I do HTTP tunneling?
- Does WebLogic JMS support SSL?
- How do I integrate non-WebLogic JMS providers with WLS?

Transaction Support

- How do two-phase or global transactions relate to WebLogic JMS?
- Why is my WebLogic JMS work not part of a user transaction (i.e., called within a transaction but not rolled back appropriately)? How do I track down transaction problems?
- How can an application do a JMS operation and have it succeed, independent of the result of the transaction?
- What happens if `acknowledge()` is called within a transaction?
- Why do I get an error when using a non-transacted `TopicSession` from a transaction-required EJB?

-
- Can I use a one-phase commit if my WebLogic JMS JDBC store is on the same database for which I am doing other database work?
 - How do I integrate an XAResource with WLS to get JTA transactions with another resource manager?
 - Why do I get an exception when I start up WebLogic JMS using an XA driver or with a TX data source?
 - Is WL JMS XAResource compliant?
 - Why can't I receive a message that I send within a container-managed transaction?
 - What happens to a message that is rolled back or recovered?

JMS Programming Practices

- Is it possible to set aside a message and acknowledge it later?
- How should I use sorted queues?
- How does sorting on message priority work?
- How do I deal with a listener that doesn't keep up with messages being sent?
- How do I get a thread dump to help track down a problem?
- Do client identifiers need to be unique?
- Are messages passed by copy/value or reference?
- How do I manage a queue to view and delete specific messages?
- How do I close a queue so that the messages will not be reloaded at the next server startup?
- Why does an Object Message print as null after it is received?
- In what order are messages delivered to a consumer?
- Why do I get an exception when trying to find a connection factory?
- Why should I avoid using message selectors?

- Is it possible to have multiple queue receivers listening on the same queue, using message selectors (typically filtering on a correlation ID) to determine which listener actually receives the message?
- Is there a way to make a queue such that if one application has one object as listener on that queue, no other application can listen to the messages on that queue?
- Why doesn't setting values work using `javax.jms.Message.setJMSPriority`, `DeliveryMode`, `Destination`, `TimeStamp` or `Expiration`?
- What care must be taken when multi-threading WebLogic JMS clients?
- How should an application be set up to subscribe to multiple topics?
- How should I use blocking and asynchronous `receive()` calls?
- What precautions should I take when I use blocking `receive()` calls?
- What is the `NO_ACKNOWLEDGE` acknowledge mode used for?
- When should I use multicast subscribers?
- When should I use server session pools and connection consumers?
- How do I issue the `close()` method within an `onMessage()` method call and what are the semantics of the `close()` method?
- How do I publish an XML message?
- How do I use WebLogic JMS in an applet?
- How do I use a startup class to initialize and later reference WebLogic JMS objects?
- Is it possible to send or receive a message from within a message listener?
- How do I create a producer pool?
- What are pending messages in the console?
- How do I use a less than or greater than on a message selector in `ejb-jar.xml`?
- Is it better to have more or fewer sessions for a given number of subscribers?
- Are foreign destinations handled within foreign JMS messages?

-
- What is the standard way to create threads, do initialization, etc. within the application server?
 - Why do I get a JNDI problem when I name a Topic A.B and a second Topic A.B.C?
 - How many messages are sent across the network for processing topic messages?
 - What should an XPATH selector look like?
 - How do I handle request/response using WebLogic JMS?
 - How do I put a message back on the queue for processing?
 - Is it okay to add new sessions and subscribers to a Queue or Topic Connection once it has been started?
 - What can I do when I get java.lang.OutOfMemoryError because producers are faster than consumers?
 - Why have different connection factories?
 - How should connections and sessions be allocated?
 - How does an application know if an application server goes down?
 - Is there a way to dynamically change an existing selector for a TopicConsumer using setMessageSelect(String s)?
 - How can I avoid asynchronous message deadlocks?

Message-Driven Beans

- What are the advantages of message-driven beans?
- How does concurrency work for message-driven beans?
- Can an MDB be a message producer or both a producer and consumer?
- If an MDB uses a durable subscription, will messages be accumulated if the MDB is not deployed?
- How do I use non-WebLogic JMS provider destinations to drive MDBs?
- Can you use a foreign JMS provider to drive an MDB transactionally?
- How do I use JTA transactions within an MDB?

- [How do server session pools and message driven beans compare?](#)

Q. What makes WebLogic JMS unique?

A. The following features make WebLogic JMS unique:

- Strict adherence to the [Sun Microsystems' JMS specification](#).
- Solid reliability, scalability, and performance.
- Integration with the application server, WebLogic Server.
- Clustering support (two-tier or three-tier). In this case, the destination location is transparent to client.
- In a clustered environment, configuration of multiple physical destinations as members of a single distributed destination set.
- By using distributed destinations, WebLogic JMS can also spread or balance the messaging load across multiple physical destinations.
- Two-phase transactions allowing interoperability between JMS applications and other resource managers (primarily databases). JMS applications can participate in transactions with other Java APIs that use Java Transaction API (JTA).
- File-based or JDBC-based persistent message storage.
- Message driven beans.
- Multicasting support allowing the delivery of messages to a select group of hosts using an IP multicast address.
- `NO_ACKNOWLEDGE` acknowledge mode.
- Extensible Markup Language (XML) messages.
- Optional message quotas on a server or destination basis.
- Multiple queue sorting options.
- Message flow control options for instructing producers to limit their message flow.
- Message paging options for swapping out messages from virtual memory to persistent storage when message loads reach a specified threshold.

-
- Messaging bridge for transferring messages between any two JMS providers—including separate implementations of WebLogic JMS.
 - Dependable customer support.

Q. Which JMS features are new in WebLogic Server 7.0?

A. The following features are new in version 7.0.

- High Availability Enhancement – WebLogic JMS takes advantage of the migration framework implemented in the WebLogic Server core for clustered environments. This allows WebLogic JMS to properly respond to migration requests and bring a JMS server online and offline in an orderly fashion. This includes both scheduled migrations as well as migrations in response to a WebLogic Server failure.

For more information, see “[Managing JMS](#)” in the *Programming WebLogic JMS* guide.

- Distributed Destinations within WebLogic Clusters – The highly available implementation of WebLogic JMS offers a level of service continuity in the event of a single server failure by enabling you to configure multiple physical destinations as members of a single distributed destination set. Specifically, an administrator can configure multiple instances of a given destination within a cluster. If one instance within the cluster fails, then other instances of the same destination will be able to provide service to JMS producers and consumers.

For more information, see “[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS* and “[Managing JMS](#)” in the *Administration Guide*.

- Flow Control – Using the Flow Control feature, you can enable a JMS server or destination to slow down message producers when it determines that it is becoming overloaded. Specifically, when a JMS server/destination exceeds its specified bytes or messages thresholds, it instructs producers to limit their message flow.

For more information, see “[Managing JMS](#)” in the *Administration Guide*.

- Messaging Bridge – A messaging bridge (also known as a *JMS bridge*) is responsible for transferring messages between two JMS providers. The WebLogic Messaging Bridge feature allows you to configure a store and forward mechanism between any two messaging providers—including separate implementations of WebLogic JMS.

For more information, see “[Using the WebLogic Messaging Bridge](#)” in the *Administration Guide*.

- Message Paging – Made available in version 6.1 Service Pack 2, this feature can free up valuable virtual memory during peak message load periods by swapping out messages from virtual memory to persistent storage when message loads reach a specified threshold. From a performance perspective, this feature can greatly benefit WebLogic Server implementations with the large message spaces that are required by today's enterprise applications.

For more information, see “[Managing JMS](#)” in the *Administration Guide*.

- Message-Driven Beans – In WebLogic Server 7.0, you can deploy an MDB that supports container-managed transactions against a foreign JMS provider. See the white paper, “Using Foreign JMS Providers with WebLogic Server” ([jmsproviders.pdf](#)) on <http://dev2dev.bea.com/technologies/jms/index.jsp> for an example of how to configure an MDB to use a foreign provider.

Q. Where can I learn more about WebLogic JMS?

A. The following links provide more information about WebLogic JMS:

Programming WebLogic JMS

“[Managing JMS](#)” in the *Administration Guide*

[Sun Microsystems' JMS Specification](#)

BEA's [dev2dev](#) web site

[Ask BEA](#)

The WebLogic JMS “[weblogic.developer.interest.jms](#)” newsgroup available on the [BEA Newsgroup](#) server.

Q. Are there C/C++ interfaces to WebLogic JMS?

A. No, this is not supported.

- Write your own interfaces using JNI.
- Setup a Servlet that your C/C++ client calls to generate a JMS message. You should spawn multiple threads in C++ and use multiple posts to pass messages via http.

Q. Is there a smaller version of the `weblogic.jar` file for supporting clients?

A. The “WebLogic Thin Client” whitepaper (`WebLogicThinClient.zip`) on on <http://dev2dev.bea.com/technologies/jms/index.jsp> describes the current options for building thin client applications with WebLogic Server.

There is also a discussion about how to generate your own slimmed down version of `weblogic.jar` in <news://newsgroups.bea.com/3ad4ad17@newsgroups.bea.com>.

Q. How do I start WebLogic Server and configure JMS?

A. Refer to “[Managing JMS](#)” in the Administration Guide for detailed instructions on starting WebLogic Server, accessing the Administration Console, and configuring Weblogic JMS.

Q. How do I configure WebLogic JMS security?

A. In previous releases of WebLogic Server, ACLs were used to protect WebLogic resources. In WebLogic Server version 7.0, security policies answer the question "who has access" to a WebLogic resource. A security policy is created when you define an association between a WebLogic resource and a user, group, or role. A WebLogic resource has no protection until you assign it a security policy.

For instructions on how to set up security for all WebLogic Server resources, see “[Setting Protections for WebLogic Resources](#)” in the *Administration Console Online Help*.

Q. Can I still use the default connection factories supported in WebLogic JMS 5.1?

A. Yes. For detailed information about using 5.1 connection factories in later versions of WebLogic JMS, see “[Porting WebLogic JMS Applications](#)” in *Programming WebLogic JMS*.

Q. Why does `JMSSESSION.createTopic` or `JMSSESSION.createQueue` fail to create a destination in WebLogic JMS 7.0? (It worked in version 5.1?)

A. For a detailed explanation of this issue, refer to the “[JMS FAQ](#)” in the version 6.1 *Frequently Asked Questions*.

Q. How do I programmatically get a list of Queues or Topics?

A. The following program uses MBeans:

```
import weblogic.management.*;
import weblogic.management.configuration.*;

InitialContext ic = new InitialContext();
MBeanHome home = (MBeanHome)ic.lookup(MBeanHome.ADMIN_JNDI_NAME);

for(Iterator i = o.getMBeansByType("JMSTopic").iterator();
i.hasNext(); ){
    WebLogicMBean wmb = (WebLogicMBean)i.next();
    System.out.println("topic name found: " + wmb.getName());
}

for(Iterator i = o.getMBeansByType("JMSQueue").iterator();
i.hasNext(); ){
    WebLogicMBean wmb = (WebLogicMBean)i.next();
    System.out.println("queue name found: " + wmb.getName());
}
```

Q. How do I use a temporary destination?

A. You must create a template on every `JMSServer` where you want to be able to create temporary destinations. You can specify multiple `JMSServer` entries to support Temporary Template and the system will load balance among those `JMSServers` to setup the temporary destination. See *How do I start WLS and configure JMS?* for a description about how to configure JMS. The resulting template definition looks something like the following:

```
<JMSTemplate Name="MyTemplate"/>
```

The `JMSServer` is defined something like:

```
<JMSServer Name="MyJMSServer" TemporaryTemplate="MyTemplate"
Targets="MyServer" >
```

After the template name, you can set any queue/topic attribute you want in the template (not including a JNDI name or topic multicast settings). The template is at the outer most level; that is, it should not be nested in your `<JMSServer>`.

Temporary destinations can only be consumed by the creating connection. Using topics, you create your temporary topic and subscribe to that temporary topic. If you want someone to publish to that temporary topic, you need to tell that someone what your topic is. You can send them a message and include your temporary topic in the `JMSReplyTo` field. The creator of the `TemporaryTopic` and the subscriber must be one in the same.

```
import javax.jms.TopicSession;
TemporaryTopic myTopic = mySession.createTemporaryTopic();
TopicSubscriber = mySession.createSubscriber(myTopic);
```

Temporary topics do not get names and cannot be subscribed to by other connections. When you create a temporary topic, the JMS provider returns a `javax.jms.Topic`. You then need to advertise that topic to other parties (those who want to publish to the topic), putting it in your `JMSReplyTo` field so that they can respond. In general, no one else can subscribe to the topic. You advertise the topic any way you want. Topics are `Serializable` (or, in our case, `Externalizable`), which allows you to pass them around in RMI calls, through a file, binding it to a name in JNDI, etc. In short, create the topic at the subscriber side and advertise so that others can publish. You can get multiple subscribers on the same connection and get concurrent processing using multiple sessions.

Q. How do I use MBeans to print runtime statistics?

A. Newsgroup article

news://newsgroups.bea.com/3B3B77A9.CDCE3954@not.my.address.com contains a program to print JMS statistics based on runtime MBeans.

Q. Can two JMS servers share the same persistent store?

A. No. Each JMS server must have its own unique persistent store. Two file-based JMS persistent stores may share the same directory, but their messages will be stored in different files. In this case, the filenames will contain different prefixes.

Two JDBC-based JMS persistent stores may share the same database, but they must be configured to use a different Prefix Name which will be prepended to the database tables. For more information on configuring the JDBC Prefix Name, see "JMS JDBC Stores" in the Administration Console Online Help. If they are configured with the same Prefix Name, persistent messages will be corrupted and/or lost.

Q. Which types of JDBC databases does WebLogic JMS support?

A. The JMS database can be any database that is accessible through a JDBC driver. WebLogic JMS detects some drivers for the following databases:

- Pointbase
- Microsoft SQL (MSSQL) Server
- Oracle

- Sybase
- Cloudscape
- Informix
- IBM DB2
- Times Ten

The `weblogic/jms/ddl` directory within the `weblogic.jar` file contains JMS DDL files for these databases, which are actually text files containing the SQL commands that create the JMS database tables.

Q. How do I use a third-party JDBC driver with WebLogic JMS?

If your JDBC driver is not included in the list of drivers in the question about JDBC databases supported by WebLogic JMS, then the tables required by JMS must be created manually.

Note: WebLogic Server only guarantees support for the JDBC drivers included in the previous list. Support for any other JDBC driver is not guaranteed.

The `.ddl` files located in the `weblogic/jms/ddl` directory of the `weblogic.jar` file may be used as templates. Use the `jar` utility supplied with the JDK to extract them to the `weblogic/jms/ddl` directory using the following command:

```
jar xf weblogic.jar weblogic/jms/ddl
```

Note: If you omit the second parameter (`weblogic/jms/ddl`), the entire `jar` file is extracted.

Follow the procedures in [JDBC Database Utility](#) in *Programming WebLogic JMS* to manually create the database tables for the JDBC store.

Another option is to consider using a file store instead of a JDBC store. File stores are easier to configure and may provide significantly better performance.

Q. What if my JDBC database becomes corrupt?

The procedures for removing and regenerating the JDBC store tables or creating the database tables manually are described in detail in [JDBC Database Utility](#) in *Programming WebLogic JMS*.

Q. How do I use persistence?

A. Use the following guidelines:

1. Make sure the JMS Server you are using has a store configured. The JMS Server configuration entry in the `config.xml` file should contain a line of the form

```
Store="<YOUR-STORE-NAME>"
```

Note that if JMS boots without a store configured, it is assumed the customer did not want one, and persistent messages are silently downgraded to non-persistent (as specified for JMS 1.0.2b).

2. Make sure you are not using "Message.setJMSDeliveryMode". This is overwritten, as it is a vendor-only method.

3. Make sure you are calling either:

```
QueueSender.send(msg, deliveryMode, ...)
```

-- or --

```
QueueSender.setDeliveryMode(deliveryMode)
```

-- or --

set `DefaultDeliveryMode mode` on connection factory in the `config.xml` file to persistent (the `QueueSender.setDeliver/send` overrides this value).

Similarly, for topics, you would set this via the `TopicPublisher`.

4. Make sure you don't have "DeliveryModeOverride" set to Non-Persistent on the Destination in the `config.xml` file.
5. If you are using `pub/sub`, only durable subscriptions persist messages. Non-durable subscriptions have no need to persist messages, as by definition they only exist for the life of the server.
6. If you are using JDBC, the JDBC tables, `JMSSTATE` and `JMSSTORE`, are created automatically when the JMS server boots. The DDL files used to create the tables are stored in `weblogic.jar` in `weblogic/jms/ddl`. The example configuration below shows a JDBC store for Oracle (refer to the [Certifications page](http://e-docs.bea.com/platform/suppconfigs/index.html) at <http://e-docs.bea.com/platform/suppconfigs/index.html> for the latest Oracle certification information). To manually create the tables (also deleting any existing tables), run `java utils.Schema` as described in the previous question.

See the question, "How do I start WLS and configure JMS?" for a description of how to configure JMS.

Here is a sample `config.xml` file resulting from configuring JMS. It should look similar to yours. If you want JMS to use a file store instead of a database, just change `JDBCStore` to `FileStore` in the `JMSServer` section.

```
<Server Name="myserver"
ListenPort="7001" DefaultProtocol="t3"
ThreadPoolSize="8" >
</Server>
<Security Realm="defaultRealm"
GuestDisabled="false" />
<Realm Name="defaultRealm"
FileRealm="defaultFileRealm" />
<FileRealm Name="defaultFileRealm"
/>
<JMSServer Name="TestJMSServer"
TemporaryTemplate="TestTemplatel"
Targets="myserver" Store="JDBCStore">
<JMSQueue Name="TestQueue1"
JNDIName="jms.queue.TestQueue1"
Template="TestTemplatel"
/>
</JMSServer>
<JMSTemplate Name="TestTemplatel"
/>
<JMSFileStore Name="FileStore"
Directory="myfilestore"
JMSServer="TestJMSServer"
/>
<JMSJDBCStore Name="JDBCStore"
ConnectionPool="testpool2"
JMSServer="TestJMSServer"
/>
<JDBCConnectionPool Name="testpool2"
Targets="myserver"
URL="jdbc:weblogic:oracle"
DriverName="weblogic.jdbc.oci.Driver"
InitialCapacity="0"
MaxCapacity="1"
CapacityIncrement="1"
Properties="user=SCOTT;password=tiger;server=bay816"
/>
</Domain>
```

The following is a sample class that sends a Topic message on construction:

```
import javax.naming.*;
import javax.jms.*;
import java.util.Hashtable;
```

```

public class t
{
public final static String DESTINATION="jms.topic.TestTopic1";

private TopicConnectionFactory connectionFactory;
private TopicConnection      connection;
private TopicSession          session;
private TopicPublisher        producer;
private TextMessage           message;
private Topic                  destination;

public t()
{
try {
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY,
"weblogic.jndi.WLInitialContextFactory");
env.put(Context.PROVIDER_URL, "t3://localhost:7001");
InitialContext ctx = new InitialContext(env);
destination = (Topic) ctx.lookup(DESTINATION);
connectionFactory = (TopicConnectionFactory)
ctx.lookup("javax.jms.TopicConnectionFactory");

connection = (TopicConnection)
connectionFactory.createTopicConnection();
session = (TopicSession) connection.createTopicSession(false,
Session.AUTO_ACKNOWLEDGE);

producer = (TopicPublisher) session.createPublisher(destination);
producer.setDeliveryMode(DeliveryMode.PERSISTENT);
message = (TextMessage) session.createTextMessage();
message.setText("hello world");
producer.publish(message);
} catch (Exception e) {
}
}
}

```

Q. How does a file store compare with a JDBC store?

A. The following are some similarities and differences between file stores and JDBC stores:

- Both have the same transaction semantics, including rolling back transactions (e.g., received messages are put back on the queue).
- Both have the same application interface (no difference in application code).

- The file store should be much faster.
- JDBC may make it easier to handle failure recovery since the JDBC interface can access the database from any client machine; with the file store, the disk must be shared or migrated.
- File store reliability is limited to reliability of disk and O/S; run it on Veritas or a RAID 5 system. Database reliability may be higher.
- File stores will require more memory, but perhaps not significantly more; it depends on how fragmented the file store gets, if the application works roughly
- FIFO it shouldn't get very fragmented at all.
- File stores generate no additional network traffic, database stores do if the database server is on a different JVM or machine.

Q. How do the JMS server/destination message maximum and threshold values work?

A. The byte and message maximum values are quotas - not flow control. Message quotas prevent a WebLogic JMS server from filling up with messages and possibly running out of memory, causing unexpected results. When you reach your quota, JMS prevents further sends with a `ResourceAllocationException` (rather than blocking). You can set quotas on individual destinations or on a server as a whole.

The thresholds are also not flow control - though they would be better suited to that application than the quotas. The thresholds are simply settings that when exceeded cause a message to be logged to the console to let you know that you are falling behind.

Note: WebLogic JMS 7.0 has a flow control feature that enables a JMS server or destination to slow down message producers when it is becoming overloaded. Specifically, when a JMS server/destination exceeds its specified bytes or messages thresholds, it instructs producers to limit their message flow. For more information, see “[Managing JMS](#)” in the *Administration Guide*.

Note that the messages maximum setting on a connection factory is not a quota. This specifies the maximum numbers of outstanding messages that can exist after they have been pushed from the server but before an asynchronous consumer has seen them; it defaults to a value of 10.

Q. How do I configure JDBC so that the JMS JDBC Store recovers automatically?

A. If you are using a JDBC store when the DBMS goes down and then comes back online, JMS cannot access the store until WebLogic Server is shut down and restarted. To work around this problem, configure the following attributes on the JDBC connection pool associated with the JMSJDBCStore:

```
TestConnectionsOnReserve="true"\  
TestTableName="[[[catalog.]schema.]prefix]JMSState"
```

Q. What is the value of WebLogic JMS clustering in version 7.0?

A. In version 6.x, you could establish cluster-wide, transparent access to destinations from any server in the cluster by configuring multiple connection factories and using *targets* to assign them to WebLogic Servers, as described in “[Managing JMS](#)” in the *Programming WebLogic JMS*. Each connection factory can be deployed on multiple WebLogic Servers. The administrator could configure multiple JMS servers on the various nodes in the cluster—as long as the servers are uniquely named—and can then assign destinations to the various JMS servers.

For WebLogic JMS 7.0, the administrator can also configure multiple destinations as part of a single distributed destination set within a cluster. Producers and consumers are able to send and receive to the distributed destination. In the event of a single server failure within the cluster, WebLogic JMS then distributes the load across all available physical destinations within the distributed destination. For more information, see “[Configuring Distributed Destinations](#)” in the *Administration Guide*.

WebLogic JMS 7.0 also takes advantage of the migration framework implemented in the WebLogic Server core for clustered environments. This allows WebLogic JMS to properly respond to migration requests and bring a JMS server online and offline in an orderly fashion. This includes both scheduled migrations as well as migrations in response to a WebLogic Server failure. For more information, see “[Managing JMS](#)” in the *Programming WebLogic JMS*.

Q. How can I control on which WebLogic Server(s) my application will run?

A. A system administrator can specify on which WebLogic Server(s) applications will run by specifying targets when configuring connection factories. Each connection factory can be deployed on multiple WebLogic servers.

Note: If you use the default connection factory, you have no control over the WebLogic Server on which the connection factory may be deployed. If you would like to target a particular WebLogic Server, create a new connection factory and specify the appropriate JMS server target(s).

Q. How do I perform a manual fail-over?

A. The procedures for recovering from a WebLogic Server failure, and performing a manual failover, including programming considerations, are described in detail in [Managing JMS](#) in the *Administration Guide*.

Q. Does the WebLogic JMS server find out about closed or lost connections, crashes, and other problems and does it recover from them?

A. Yes, provided the client closes its resources, the network connection fails, or the JVM it's in exits, even for multicast pub/sub. It detects losses of "peers" and recovers.

Q. Do I need to use the WLS T3 protocol?

A. J2EE is all about making the interfaces standard. WebLogic's implementation of the RMI specification uses a proprietary wire-protocol known as T3. Sun's reference implementation of RMI uses a proprietary protocol called JRMP. The fact is that WebLogic developed T3 because they needed a scalable, efficient protocol for building enterprise-class distributed object systems with Java.

While T3 is specific to WebLogic, your application code does not need to know anything about T3 so you should not worry about this. Externalize the "WebLogic-specific strings" (`PROVIDER_URL`, `INITIAL_CONTEXT_FACTORY`, etc.) to a properties file (or somewhere) and you can make your code completely portable to where you only need change these in the properties file to get your code to run on another J2EE application server.

Q. How do I do HTTP tunneling?

A. If you want to use HTTP tunneling (wrap every message in HTTP to get through a firewall), you need to add `TunnelingEnabled="true"` into your `<Server>` definition in the `config.xml` file or check the appropriate box on the console. Then use a URL like `http://localhost:7001` instead of `t3://localhost:7001` for `Context.PROVIDER_URL` when getting your `InitialContext`. If you want HTTP tunneling with SSL, use `https://localhost:7002` (where https uses HTTP tunneling with SSL and 7002 is the secure port that you configured). You will pay a performance penalty for doing this, so only use tunneling if you really need to (i.e., need to go through a firewall).

Q. Does WebLogic JMS support SSL?

A. Yes, SSL is supported in the WebLogic JMS implementation. It is automatically used based on using a URL starting with "t3s:" instead of "t3:" when looking up the initial JNDI context.

Q. How do I integrate non-WebLogic JMS providers with WLS?

A. Refer to the white paper, “Using Foreign JMS Providers with WebLogic Server” (`jmsproviders.pdf`) on <http://dev2dev.bea.com/technologies/jms/index.jsp> for a discussion on integrating MQ Series, IBus MessageServer, Fiorano, and SonicMQ.

Q. How do two-phase or global transactions relate to WebLogic JMS?

A. A two-phase or global transaction allows multiple resource managers (including EJBs, databases, and JMS servers) to participate in a single transaction.

For example, a client can use a two-phase transaction to send a message from a queue on one JMS server (server A) to a queue on another JMS server (server B). Each server has a unique persistent store. When the transaction is committed, the message is made visible on server B. If the transaction rolls back, the message is put back on the queue on server A.

Note: If both queues happen to be on the same JMS server, then a one-phase transaction is used.

Q. Why is my WebLogic JMS work not part of a user transaction (i.e., called within a transaction but not rolled back appropriately)? How do I track down transaction problems?

A. Usually this problem is caused by explicitly using a transacted session which ignores the external, global transaction by design (a [JMS specification](#) requirement). A transacted JMS session always has its own inner transaction. It is not affected by any transaction context that the caller may have.

It may also be caused by using a connection factory that is configured with "UserTransactionsEnabled" set to false.

1. You can check if the current thread is in a transaction by adding these two import lines:

```
import javax.transaction.*
import weblogic.transaction.*;
```

and adding the following lines (i.e., just after the begin and just before every operation).

```
Transaction tran = TxHelper.getTransaction();
System.out.println(tran);
System.out.println(TxHelper.status2String(tran.getStatus()));
```

This should give a clear idea of when new transactions are starting and when infection is occurring.

2. Ensure that the thread sending the JMS message is infected with a transaction. Check that the code is not using a transacted session by setting the first parameter of `createQueueSession` or `createTopicSession` to `false`. Note that creating the connection and/or session is orthogonal to the transaction. You can begin your transaction before or after. You need only start the transaction before you send or receive messages.
3. Check that the `UserTransactionsEnabled` flag is explicitly set to `true` for the connection factory in the `config.xml` file since the default for user-configured connection factories for this value is `false`. If you are using one of the pre-configured connection factories they are set as follows:

```
weblogic.jms.ConnectionFactory disables user transactions so
don't use this one for the case where user transactions are
desired;
```

```
javax.jms.QueueConnectionFactory and
javax.jms.TopicConnectionFactory enable user transactions.
```

4. You can trace JTA operations by starting the server with this additional property:

```
-Dweblogic.Debug.DebugJMSXA=true
```

You should see trace statements like these in the log:

```
XA ! XA(3163720,487900) <RM-isTransactional() ret=true>
```

This can be used to ensure that JMS is infected with the transaction.

Q. How can an application do a JMS operation and have it succeed, independent of the result of the transaction?

A. Basically, the JMS operation must be done using a transacted session or the transaction must be suspended/disabled as follows (pick one or more of the following).

1. Suspend the current transaction prior to making the JMS call and resume it after completing it. The code looks something like this:

```
import javax.transaction.Transaction;
import javax.transaction.TransactionManager;

TransactionManager tranManager=
TxHelper.getTransactionManager();
Transaction saveTx = null;
try {
```

```
saveTx = tranManager.suspend();
... do JMS work, it will not participate in transaction
} finally {
// must always resume suspended transactions!
if (saveTx != null) tranManager.resume(saveTx);
}
```

2. Use a transacted session by specifying true for the first parameter to `createQueueSession` or `createTopicSession`.
3. Use a connection factory with user transactions disabled. That is, check that the `UserTransactionsEnabled` flag is explicitly set to false for the connection factory in the `config.xml` file or use the default for a user-configured connection factory for this value which is false. The pre-configured connection factory `weblogic.jms.ConnectionFactory` disables user transactions.

A transacted JMS session always has its own inner transaction. It is not affected by any transaction context that the caller may have. A non-transacted JMS session is more complicated. If you use the WLS 6.x or later default factory

`weblogic.jms.ConnectionFactory`, the session does not participate in a user transaction because the `UserTransactionsEnabled` flag is set to "False". However, if you use the deprecated WebLogic 5.1 default

`javax.jms.QueueConnectionFactory` or

`javax.jms.TopicConnectionFactory` factories, or you define your own factory and set the `UserTransactionsEnabled` flag to "True", the JMS session participates in the outer transaction, if one exists and the JMS session is not transacted.

Q. What happens if `acknowledge()` is called within a transaction?

A. As per [Sun Microsystems' JMS specification](#), when you are in a transaction, the `acknowledgeMode` is ignored. If `acknowledge()` is called within a transaction, it is ignored.

Q. Why do I get an error when using a non-transacted `TopicSession` from a transaction-required EJB?

A. Whenever two resources (such as JMS and a database) participate in a transaction, the transaction becomes two-phase. The database driver you are using is not XA compatible and can't normally participate in a two-phase transaction. The solution is to either use an XA compatible driver, or to configure the `JDBCTxDataSource` value to set `enableTwoPhaseCommit` to true. The caveat for the latter is that this can lead to heuristic errors. If you don't want JMS to participate in the current transaction, see the question "How can an application do a JMS operation and have it succeed, independent of the result of the transaction?".

Q. Can I use a one-phase commit if my WebLogic JMS JDBC store is on the same database for which I am doing other database work?

A. No. WebLogic JMS is its own resource manager. That is JMS itself implements XAResource and handles the transactions without depending on the database (even when the messages are stored in the database). That means whenever you are using JMS and a database (even if it is the same database as the JMS messages are stored) then it is 2PC.

You may find it will aid performance if you ensure the connection pool used for the database work exists on the same server as the JMS queue—the transaction will still be two-phase, but it will be handled with less network overhead. Another performance boost might be achieved by using JMS file stores rather than JMS JDBC stores.

Q. How do I integrate an XAResource with WLS to get JTA transactions with another resource manager?

A. See the white paper, “Using JTA Transactions to Envelope WLS JMS and IBM MQSeries” (`jmsjta.doc`) on <http://dev2dev.bea.com/products/wlserver61/resources.jsp> for a discussion on integrating MQ Series.

Q. Why do I get an exception when I start up WebLogic JMS using an XA driver or with a TX data source?

A. You cannot use a TX data source with JMS. JMS must use a JDBC connection pool that uses a non-XAResource driver (you can't use an XA driver or a JTS driver). Do not set the `enableTwoPhaseCommit` option. JMS does the XA support above the JDBC driver.

Q. Is WL JMS XAResource compliant?

A. Yes. WebLogic JMS 7.0 fully implements the XAResource interface, as defined in the JTA specification. Version 7.0 also supports the XAConnection, XAConnectionFactory, XAQueueConnection, XAQueueConnectionFactory, XAQueueSession, XASession, XATopicConnection, XATopicConnectionFactory, and XATopicSession methods. These methods are defined as optional in [Sun Microsystems' JMS specification](#) and are not part of the XAResource interface.

The `XAResource` interface has methods like `start`, `prepare`, and `commit` that the transaction manager needs to manage a JTA transaction, and these are provided by WebLogic JMS. When you wrap a JMS send or receive with a JTA transaction (`begin/commit`), the JMS work is part of that transaction (unless you are using a transacted session).

It does mean that you can't use an `XAQueueConnectionFactory` to get an `XAQueueConnection` to get an `XAQueueSession` to call `getXAResource()`. However, WebLogic JMS is automatically registered by WebLogic JTA so you don't need to worry about that. It also means that you can't use WebLogic JMS with another transaction manager driving the transaction (at least, not with any documented interfaces).

Q. Why can't I receive a message that I send within a container-managed transaction?

A. If you are using container-managed transactions, the original message sent from the EJB will never be sent. Here is what is happening.

1. Container starts transaction.
2. Start method.
3. Generate new message.
4. Send message (message isn't sent – it's buffered until transaction commit).
5. Do a blocking receive on a queue.
6. End method.
7. Transaction Commit never Reached because original message was never sent because you can't get past blocking receive.

The solution is to either use bean-managed transactions, or to break the send and receive into two separate methods.

Q. What happens to a message that is rolled back or recovered?

A. When a message is rolled back or recovered, it is requeued. Where it goes in the queue is a function of the message relative to the sort order for the queue and the other messages in the queue.

In an unordered queue (most typical), the message would typically go at the head of the queue (since the queue is still sorted by arrival time – FIFO) unless some other message is being recovered at that same time, but still the sort order resolves any conflicts. The message is redelivered to the first available consumer as soon as possible. If there are no consumers when it is requested, then it would just sit there until there is one.

You can configure the maximum number of times that a message may be requested. You can also specify a delay time before which the message will not be available after being requested. If a message reaches its maximum request limit, it is put on an error destination if one is configured or silently dropped.

Q. Is it possible to set aside a message and acknowledge it later?

A. There are no special primitives for doing this. Here are two possible solutions.

One approach is to use multiple sessions as in the following:

```
while (true) {
    Create a session, subscribe to one message on durable
    subscription
    Close session
    Save session reference in memory
    To acknowledge the message, find the session reference and call
    acknowledge() on it.
}
```

Another solution is to use transactions and suspend the work as follows:

```
start transaction
while(true) {
    message = receive();
    if (message is one that I can handle)
        process the message
        commit
    } else {
        suspend transaction
        put transaction aside with message
        start transaction
    }
}
```

To "acknowledge" the message:

```
resume user transaction
commit
```

To "recover" the message:

```
resume user transaction
rollback
```

Each time you suspend, you need to push the transaction onto a stack or list possibly with the message so you can process it or roll it back later. This solution is high overhead in that there can be a large build up of outstanding transactions. Note that transactions have timeouts and it may rollback on its own, which means you can get the message again (in a different transaction). Note also that there are some practical limits on the number of transactions you should leave outstanding. The default limit is something like 10000. Eventually you want to go back to your stack/list and commit/rollback the transactions. Note that transaction references (`javax.transaction.Transaction`) are not Serializable.

Q. How should I use sorted queues?

A. Destination keys are used to define the sort order for a specific destination. Destination keys can be message header or property fields. For a list of valid message header and property fields, refer to the Programming WebLogic JMS.

Queues can be sorted in ascending or descending order based on the destination key. A destination is considered to be first-in-first-out if a destination key is defined as ascending for the `JMSMessageID` message header field, and last-in-first-out if defined as descending. The key defined for the `JMSMessageID` header field, if specified, must be the last key defined in the list of keys.

You can define multiple destination keys to sort a destination.

To create a destination key, use the Destination Keys node in the Administration Console. For more information, refer to the [Administration Console Online Help](#).

Q. How does sorting on message priority work?

A. First, you need to add a key to the destination (by default, they are not sorted), choosing `JMSPriority` as the key. If you want 0 to be your highest priority, make the key ascending. If you want 9 to be the highest priority, make the key descending.

Second, the priority must be set using either the producer or on the send, not the message.

Third, the priority sorting only comes into play if there are multiple messages waiting on the queue. If the receiver is always caught up with the sender, then the messages will be processed in the order in which they come in.

Q. How do I deal with a listener that doesn't keep up with messages being sent?

A. Here are a few guidelines:

- Consider using multiple listeners.
- Consider reducing the processing in the listener (for example, don't call "System.out").
- Sends are in general faster than receives. Consider lowering receive overhead by not acknowledging every message, but by deferring acknowledge until several messages have been received using `client.acknowledge`. With asynchronous listeners, the last few messages may not get acknowledged without extra code logic if they don't fall on an acknowledge "boundary" (received 9 out of 10, but don't receive the 10th).

Q. How do I get a thread dump to help track down a problem?

A. Ways to get a thread dump:

- Try running this from the command line (after running the `setEnv` script in `WL_HOME\server\bin`):

```
java weblogic.Admin -url t3://localhost:7001 THREAD_DUMP
```
- On Windows, from the console window, enter `Ctrl+Break`.
- On UNIX, signal the server using `kill -3`.

Q. Do client identifiers need to be unique?

A. Using durable subscribers, if subscribing from two different clients and using the same connection factory and the connection factory has a configured `clientID`, distinct `clientID`'s must be set for each after the `TopicConnection` has been created. Each connection needs a unique `clientID`. If you configure a connection factory with a `clientID`, you can only create one `TopicConnection` with that factory until that one connection is closed.

Q. Are messages passed by copy/value or reference?

A. Messages, from an application point of view, are passed by value (a copy is made). Internally, every effort is made to optimize message passing.

When the producer is on the same JVM as the JMS server, a copy is made of the message before it is actually placed in the destination. If a consumer is on the same JVM as the JMS server, a copy is made before it is given to the consumer. This preserves the message that is actually in the destination. Passing by reference is a violation of [Sun Microsystems' JMS specification](#). Specifically, if one person produces a message and then modifies the message in their space it should not affect the message that was already sent. As such, the message sent must be a copy. The same is true on the receive side. If someone consumes the message and then does something like a clear properties, sets a header field or makes any modification of any kind, that change cannot affect the messages received by other consumers. So, the messages given to receivers on the same JVM are also copies (one for each receiver). Also, if you pass a reference to a receiver and the receiver modifies the message and then does a rollback or a recover, the message in the destination would be modified for the person who eventually does receive it. Then it appears that the message sent is not the message received. It has been modified by someone who received it but then put it back.

Q. How do I manage a queue to view and delete specific messages?

A. Write a program that uses a `QueueBrowser`. Then delete specific messages by using a selector with the message identifier as in the following example:

```
String selector = "JMSMessageID = '" + message.getMessageID() + "'";
```

Keep in mind that the queue browser is a not a “live” view of the queue. It is a snap-shot.

Q. How do I close a queue so that the messages will not be reloaded at the next server startup?

A. JMS does not define the deletion of queues. Using `WebLogic JMS`, you can administratively create and delete queues. There is no way to dynamically delete queues at run-time.

You can acknowledge the messages or not use persistence. Further, you can use temporary queues. They exist only for the life of the connection. You can also use durable subscriptions. A durable subscriber’s message gets deleted if that durable subscriber unsubscribes, but the topic continues to exist.

Q. Why does an Object Message print as null after it is received?

A. The object isn't deserialized until `ObjectMessage.getObject()` is called. The `toString()` will continue to print null until this occurs. `WebLogic JMS 6.x` or later automatically deserializes the message if the application has not called `setObject()`.

Q. In what order are messages delivered to a consumer?

A. Order is maintained between any producer and consumer for like delivery mode, sort order, and selector in the absence of a rollback or recover. There are no guarantees of order when multiple producers send to a single consumer or multiple consumers receive from multiple producers.

Order is generally maintained between a producer and a consumer. However, non-persistent messages can get ahead of persistent messages of a higher sort order (i.e., higher priority), can move ahead of each other and a recover or rollback puts messages that were already received back into the queue/topic, which affects order.

Most messaging systems (including WebLogic JMS) maintain order between a producer and a destination and then order between the destination and the consumer. So, once things arrive at the destination, the order does not change.

Finally, the asynchronous pipeline that is supported in WebLogic JMS affects the ordering. By default there can be as many as ten outstanding messages pushed out from the server to an asynchronous client that have not been seen by the client yet. If the asynchronous consumer is "caught" up, these messages will not be sorted. Destination sorting does not occur in the pipeline. If a destination is sorted by priority, and a new message comes in of higher priority than those messages already in the pipeline, it will not leap ahead in the pipeline, it will become first in the destination. The size of the pipeline is configurable; see the `MessagesMaximum` setting on the connection factory used. If you want real priority sorting, change the maximum number of messages on the factory to one.

Q. Why do I get an exception when trying to find a connection factory?

A. The exception is usually something like `java.io.InvalidClassException` or `java.lang.NoClassDefFoundError`.

Make sure `weblogic.jar` is in the `CLASSPATH` of the client. Also make sure you have the correct Java run-time jar files included (i.e., you might need `rt.jar`).

Q. Why should I avoid using message selectors?

A. We recommend avoiding an architecture that requires heavy use of selectors. If you need to use selectors, try to use topic selectors. They only incur a charge once per message per subscriber. If the message doesn't match a subscriber's selection criteria, it is simply ignored and not placed in that subscriber's "subscription".

For Queues, selectors can be much more heavy-weight. Each receive request can incur a scan of the entire queue in a search for a message that matches the receiver's selection criteria. This can get really bad if the selection criteria are restrictive and the queue depth is large.

The topic generally performs better. However, it depends on the message mix and queue depth. When using queues, if there are a lot of messages that don't match what you are looking for, you are going to compare your selector against those messages over and over again each time you go back to look for one of yours. When using a topic, each message is compared against all consumers exactly once. There is never any duplicate work.

Also, if you are using synchronous queuing (posting a receive call), each time you go you have to look for a message. That is, you wait during the time the server is comparing. When you use topics, the message has been compared up front and essentially put in your own private queue. You don't have to look at that point. It is FIFO from your own private queue. The server can do the comparisons while you are processing orders as opposed to your application blocking while the server finds you a message.

Q. Is it possible to have multiple queue receivers listening on the same queue, using message selectors (typically filtering on a correlation ID) to determine which listener actually receives the message?

A. [Sun Microsystems' JMS specification](#) does not define the behavior. WebLogic JMS 6.x or later supports this. Whenever a message lands on the queue, JMS will look at all of the consumers for the queue in the order in which they did their receives; whoever is the first to match the message receives the message. For asynchronous queue consumers, you must set your listener first, and then you will be first in the consumer list. However, each time an asynchronous queue consumer receives a message, the consumer goes to the end of the list. Note that unlike topics, if a message does not match a selector, it is left in the queue until someone has no selector, or a selector that does match.

Q. Is there a way to make a queue such that if one application has one object as listener on that queue, no other application can listen to the messages on that queue?

A. No. An alternative is to create a topic with a single durable subscription because a durable subscription may only have one consumer associated with it. The only drawback is that selectors would no longer work the same as they do with queues. Changing the selector on a durable subscription "resets" the subscription as per [Sun Microsystems' JMS specification](#), causing all messages currently in the subscription to be deleted.

Q. Why doesn't setting values work using `javax.jms.Message.setJMSPriority`, `DeliveryMode`, `Destination`, `TimeStamp` or `Expiration`?

A. These methods are for vendor use only. The message values are overwritten on each `send/publish`. You should use the equivalent methods on the `MessageProducer`, `QueueSender`, or `TopicPublisher` to set these values (i.e., `setJMSPriority`, `setDeliveryMode`, `setTimeToLive`). Check to see that these values are not being overridden by the optional template configuration override values.

Q. What care must be taken when multi-threading WebLogic JMS clients?

A. The rules for multi-threading are described in section 2.8 of the [JMS specification](#), with additional language in sections 4.4.6 on session usage, 4.4.9 on using multiple sessions, and 4.4.17 on concurrent message delivery. In a nutshell, it states that JMS sessions are single-threaded. Consequently, if multiple threads simultaneously access a session or one of its consumers or producers the resulting behavior is undefined. In addition, if multiple asynchronous consumers exist on a session, messages will be delivered to them in series and not in parallel.

To take advantage of multiple threads with JMS, use multiple sessions. For example, to allow parallel synchronous receive requests, design the application so that only one consumer may be active per session and use multiple sessions.

Q. How should an application be set up to subscribe to multiple topics?

A. If you want to listen to `N` topics, using `N` subscribers and `N` sessions gives you concurrency up to `N` simultaneous threads of execution provided you have that many threads to work with. `N` subscribers and 1 session serializes all subscribers through that one session. If the load is heavy they may not be able to keep up without the extra threads. Also, if you are using `CLIENT_ACKNOWLEDGE`, `N` sessions gives you `N` separate message streams that can be individually recovered. Having 1 session crosses the streams giving you less control.

As of version 6.x or later, WebLogic JMS on the server side efficiently uses a small, fixed number of threads independent of how many client sessions there are.

Q. How should I use blocking and asynchronous `receive()` calls?

A. The synchronous `receive()` method blocks until a message is produced, the timeout value, if specified, elapses or the application is closed. We strongly recommend that you avoid using blocking `receive()` calls on the server side because a synchronous `receive()` call consumes resources for the entire duration that the call is blocked.

Server deadlock might also result if receiving messages use client servlets, as described in the section “[Receiving Messages with Client Servlets](#)” in *Programming WebLogic JMS*.

When methods are received asynchronously, the application is notified using a message listener only when a message has been produced, so no resources are consumed waiting for a message.

Q. What precautions should I take when I use blocking `receive()` calls?

A. If your application design requires messages to be received synchronously, we recommend using one of the following methods listed in order of preference:

- Pass a timeout value as an argument to the `receive()` method and set it to the minimum value greater than zero, that is allowed by the application to avoid consuming threads that are waiting for a response from the server.
- Use the `receiveNowait()` method which returns the next message or a null value if no message is currently available. In this case, the call does not block. The servlet should provide a way to return to or reschedule the request, without calling `wait()`.

Note: Use of this option should be minimized, as it may deadlock a busy server.

- Ensure that more threads are configured than the number of possible simultaneous blocking `receive()` calls.

Q. What is the `NO_ACKNOWLEDGE` acknowledge mode used for?

A. The `NO_ACKNOWLEDGE` acknowledge mode indicates that received messages do not need to be specifically acknowledged which improves performance, but risks that messages are lost. This mode is supported for applications that do not require the quality of service provided by session acknowledge and that do not want to incur the associated overhead.

Messages sent to a `NO_ACKNOWLEDGE` session are immediately deleted from the server. Messages received in this mode are not recovered and, as a result, messages may be lost and/or duplicate message may be delivered if an initial attempt to deliver a message fails.

Note: You should avoid using this mode if your application cannot handle lost or duplicate messages. Duplicate messages may be sent if an initial attempt to deliver a message fails.

In addition, we do not recommend that this acknowledge mode be used with persistent messaging, as it implies a quality of service that may be too low for persistent messaging to be useful.

Q. When should I use multicast subscribers?

A. Multicasting enables the delivery of messages to a select group of hosts that subsequently forwards the messages to multicast subscribers. The benefits of multicasting include:

- Near real-time delivery of messages to host group.
- High scalability due to the reduction in the amount of resources required by the JMS server to deliver messages to multicast subscribers.

Note: Multicasting is only supported for the Pub/sub messaging model.

For an example of when multicasting might be useful, consider a stock ticker. When accessing stock quotes, timely delivery is more important than reliability. When accessing the stock information in real-time, if all, or a portion, of the contents is not delivered, the client can simply request the information be resent. Clients would not want to have the information recovered in this case because by the time it is redelivered it would be out-of-date.

Multicast messages are not guaranteed to be delivered to all members of the host group. For messages requiring reliable delivery and recovery, you should not use multicasting.

Q. When should I use server session pools and connection consumers?

A. WebLogic JMS implements an optional JMS facility for defining a server-managed pool of server sessions. This facility enables an application to process messages concurrently. A `ConnectionConsumer` object uses a server session to process received messages. If message traffic is heavy, the connection consumer can load each server session with multiple messages to minimize thread context switching. Multiple connection consumers can share server sessions in a server session pool.

To learn how to use the connection consumers within an application, see the section [Processing Messages Concurrently](#) in *Programming WebLogic JMS*, or the `javax.jms.ConnectionConsumer` javadoc.

Note: Server session pools can also be implemented using Message Driven Beans. Using MDBs is preferable to using server session pools; this issue is addressed in, "[How do server session pools and message driven beans compare?](#)" For information on using message driven beans to implement server session pools, see [Programming WebLogic Enterprise JavaBeans](#).

Q. How do I issue the `close()` method within an `onMessage()` method call and what are the semantics of the `close()` method?

A. If you wish to issue the `close()` method within an `onMessage()` method call, the system administrator must select the Allow Close In OnMessage check box when configuring the connection factory. For more information, see [JMS Connection Factories](#) in the *Administration Console Online Help* guide. If this check box is not selected and you issue the `close()` method within an `onMessage()` method call, the call will hang.

The `close()` method performs the following steps to execute an orderly shutdown:

- Terminates the receipt of all pending messages. Applications may return a message or null if a message was not available at the time of the close.
- Waits until all message listeners that are currently processing messages have completed (except for the message listener from which the `close()` method is being called).
- Rolls back in-process transactions on its transacted sessions (unless such transactions are part of an external JTA user transaction).
- Does not force an acknowledge of client-acknowledged sessions. By not forcing an acknowledge, no messages are lost for queues and durable subscriptions that require reliable processing.

When you close a connection, all associated objects are also closed. You can continue to use the message objects created or received via the connection, except the received message's `acknowledge()` method. Closing a closed connection has no effect.

Note: Attempting to acknowledge a received message from a closed connection's session throws an `IllegalStateException`.

When you close a session, all associated producers and consumers are also closed.

For more information about the impact of the `close()` method for each object, see the appropriate `javax.jms` javadoc.

Q. How do I publish an XML message?

A. Follow these steps:

1. Generate XML from the DOM document tree.
2. Serialize the generated DOM document to a `StringWriter`.
3. Call `toString` on the `StringWriter` and pass it into `message.setText`.
4. Publish the message.

Q. How do I use WebLogic JMS in an applet?

A. This topic is covered in news://newsgroups.bea.com/3ad321d5@newsgroups.bea.com.

Q. How do I use a startup class to initialize and later reference WebLogic JMS objects?

A. This topic is covered in news://newsgroups.bea.com/3ad0d7f3@newsgroups.bea.com. The sample code does not cleanup properly at shutdown. You can use a shutdown class that does something like the following:

```
JMSObject WLSobject = null;
try {
    WLSobject = JMSStartup.getJMSObject();
    WLSobject.JMSCleanup();
} catch(Exception e) {}
```

Servlets can provide a nice solution to provide both initialization and cleanup. See the question in this section "What is the standard way for creating threads, doing initialization, etc. within the application server?".

Q. Is it possible to send or receive a message from within a message listener?

A. Yes. You can send to or receive from any queue or topic from within in a message listener.

If it's not an MDB, you can use the same `Connection` or `Session` that the `onMessage()` is part of to do this. When you create your message listener, you pass in a session in your constructor. Then you have access to the session in your `onMessage` method and you would be able to make synchronous, not asynchronous, calls from within the

`onMessage` method. Do not use another Session that is servicing another `onMessage()` because that would multi-thread that Session and Sessions don't support multi-threading.

When things are done non-transactionally, there can be duplicates or lost messages (assuming your `onMessage()` code is attempting to forward messages):

1. If you call `acknowledge` after the `publish()` and the `acknowledge` fails for whatever reason (network/server failure), then you will see the message again and will end up publishing twice (possible duplicate semantics). You can try to keep track of sequence numbers to detect duplicates but this is not easy.
2. If you call `acknowledge` before the `publish()`, you get at-most-once semantics. If the `publish()` fails, you don't know if the failure occurred before or after the message reached the server.

If you want exactly once, transactional semantics using `onMessage`, you must use transactional MDBs. The `onMessage()` for a transactional MDB starts the transaction, includes the WebLogic Server JMS message received within that transaction and the `publish()` would also be in the same transaction. The following code sends a response to each message that it receives. It creates the connection, etc. in the `ejbCreate` method so that it doesn't need to create it every time `onMessage` is called. The `QueueSender` is anonymous (null `Queue`) since we don't know to whom we will have to reply. The `ejbRemove` method cleans up by closing the connection. This same approach can be used to create a receiver, subscriber or publisher.

```
import javax.ejb.CreateException;
import javax.ejb.EJBContext;
import javax.naming.*;
import javax.naming.directory.*;
import java.util.Hashtable;
import javax.ejb.MessageDrivenBean;
import javax.ejb.MessageDrivenContext;
import javax.jms.*;

public class MDB
implements MessageDrivenBean, MessageListener {
    public static final String WLSqcf =
    "javax.jms.QueueConnectionFactory";
    public static final String WLSqname =
    "jms.queue.TestQueue1";
    public static final String WLSurl =
    "t3://localhost:7001";
    public static final String WLSJNDIfactory =
    "weblogic.jndi.WLInitialContextFactory";
    private MessageDrivenContext context;
```

```
private QueueSession session;
private QueueConnection connection = null;
private QueueConnectionFactory factory;
private InitialContext ctx;
private QueueSender QueueSender;

// Required - public constructor with no argument
public MDB() {}

// Required - ejbActivate
public void ejbActivate() {}
// Required - ejbRemove
public void ejbRemove() {
    context = null;
    if (connection != null) {
        try {
            connection.close();
        } catch (Exception e) {}
        connection = null;
    }
}

// Required - ejbPassivate
public void ejbPassivate() {}

public void setMessageDrivenContext(
    MessageDrivenContext mycontext) {
    context = mycontext;
}

// Required - ejbCreate() with no arguments
public void ejbCreate () throws CreateException {
    try {
        // Get the initial context
        Hashtable env = new Hashtable();
        env.put(Context.INITIAL_CONTEXT_FACTORY, WLSJNDIFactory);
        env.put(Context.PROVIDER_URL, WLSurl);
        env.put(Context.REFERRAL, "throw");
        ctx = new InitialContext(env);

        factory = (QueueConnectionFactory) ctx.lookup(WLSqcf);

        // Create a QueueConnection, QueueSession, QueueSender
        connection = factory.createQueueConnection();
        session = connection.createQueueSession(false,
            Session.AUTO_ACKNOWLEDGE);
        queueSender = session.createSender(null);
        connection.start();
    } catch (Exception e) {
        throw(new CreateException(e.toString()));
    }
}
```

```

    }
}

// Implementation of MessageListener
// Throws no exceptions
public void onMessage(Message msg) {
    try {
        System.out.println("MDB: " +
            ((TextMessage)msg).getText());
        msg.clearBody();
        ((TextMessage)msg).setText("reply message");
        queueSender.send((Queue)msg.getJMSReplyTo(), msg);
    }
    catch(Exception e) { // Catch any exception
        e.printStackTrace();
    }
}
}
}

```

This approach creates a connection per EJB/MDB instance, so you might want to create a producer pool that is shared by the EJB instances. This is done by writing a class that populates a static pool with producers (see the next question for a sample producer pool). The `onMessage` call grabs a producer when needed. Since Sessions must be single threaded, make sure there is only one producer per session within the producer pool.

Q. How do I create a producer pool?

A. The following is some pseudo-code for a producer class.

```

class ProducerPool {
    static Hashmap pSets = new Hashtable();
    static Hashmap inUse = new Hashtable();

    QueueSender get(String contextURL,
        String connectionFactoryName,
        String destinationName) {
        String lookup = contextURL+";"+"connectionFactName+";"+"destName;
        synchronized(pSets) {
            producer set = pSets.get(lookup);
            if (set != null && set not empty)
                qs = set.removeFirst();
        }
        if (producer == null) {
            create ctx
            get connect factory
            create connection
            create session
        }
    }
}

```

```
        look up destination
        qs = create queue sender
    }
    synchronized(inUse) {
        inUse.put(qs, lookup);
    }
    return qs;
}

void put(QueueSender qs) {
    String lookup;
    synchronized(inUse) {
        lookup = inUse.remove(p);
    }
    synchronized(pSets) {
        producer set = pSets.get(lookup);
        if (set == null) {
            producer set = new producer set
            pSets.put(lookup, producer set);
        }
        producer set.add(qs);
    }
}
}
```

Note: Static classes may be garbage collected if there are no references to them, so make sure the application server has a permanent pointer to them in some manner. One way is to reference it permanently from within a servlet or EJB when they are initialized at startup.

Here is an example of using the producer pool within the `onMessage` method.

```
onMessage() {
    QueueSender qs = ProducerPool.get(...);
    qs.send(...);
    ProducerPool.put(qs);
}
```

You can pre-populate this pool by calling it from a startup class or a load-on-start servlet class.

Q. What are pending messages in the console?

A. Pending means the message could have been:

- sent in a transaction but not committed.

-
- received and not acknowledged.
 - received and not committed.
 - subject to a redelivery delay (as of WebLogic JMS 6.1 or later).
 - subject to a delivery time (as of WebLogic JMS 6.1 or later).

A rolled back message remains pending until the transaction actually rolls back. Rolling it back multiple times does not cause double counting, nor does an exception that set a transaction as `rollbackOnly` followed by an actual rollback.

Current implies messages that are not pending.

Total implies total since server last started. The byte counts only consider the payload of messages which includes the properties and the body but not the header.

Q. How do I use a less than or greater than on a message selector in `ejb-jar.xml`?

A. Enclose the selector in a CDATA section. That will prevent the XML parser from thinking that less than or greater than is a tag.

```
<jms-message-selector>  
<![CDATA[ JMSXAppID <> 'user' ]]>  
</jms-message-selector>
```

Q. Is it better to have more or fewer sessions for a given number of subscribers?

A. Using N sessions for N subscribers gives you concurrency up to N simultaneous threads of execution provided you have as many threads to work with. Each Session gets its own thread as long as there are enough threads available. Otherwise, the sessions serially reuse the available threads.

One session for N subscribers serializes all subscribers through that one session. If the load is heavy they may not be able to keep up without the extra threads.

If you are using `CLIENT_ACKNOWLEDGE`, N sessions gives you N separate message streams that can be individually recovered. Having one session crosses the streams giving you less control.

Q. Are foreign destinations handled within foreign JMS messages?

A. WebLogic Server JMS does not know what to do with foreign destinations that it runs into. This issue has been discussed with Sun and the specification does not clearly define destinations well enough for vendors to interoperate at that level. They agree

that it is sufficient not to handle foreign destinations preferably in such a way that sending/receiving still work. For WebLogic JMS, if you do a `setJMSdestination` (you should not because it is only for the provider to set it) with a foreign destination, it gets ignored (set to null). Similarly, if you do a `setJMSReplyTo` for a foreign destination, WebLogic JMS will ignore it (set it to null).

Q. What is the standard way to create threads, do initialization, etc. within the application server?

A. Threads should generally not be created by the user directly because things may not work correctly. User-created threads do not have some of the thread-local variables pre-set by WebLogic when it creates its own execute threads, the associated transaction context, or the environment such as the proper class loader. The WebLogic-specific way of doing this is with a startup class or using the WebLogic Time Services. The portable way to do this is to define a load-on-startup servlet, doing the initialization in the `init()` method and the cleanup in the `destroy()` method. The servlet itself does nothing. This approach also allows for undeploy/redeploy of the application without restarting the server, including proper cleanup/initialization each time. It also provides more dynamic management of the dependent classes without restarting the server.

Q. Why do I get a JNDI problem when I name a Topic A.B and a second Topic A.B.C?

A. This is a JNDI implementation issue. JNDI uses the dots to build a directory-like structure. A given element cannot be both a node and a leaf in the tree. In this example, B is used as a leaf off of A, but then is used as a node off of which C is a leaf.

Q. How many messages are sent across the network for processing topic messages?

A. If there are three subscribers for a message, for example, one of which with a selector that does not match, how many messages are sent?

In WebLogic JMS 6.x or later, when all three consumers are in the same session, we send one copy of the message across the network for all subscribers that are not flow-controlled. Once a consumer is flow-controlled for not acknowledging enough messages, no messages are sent until flow control is eased. So, the answer is usually one, but may be two. The selection is done on the server side so a subscriber that doesn't match doesn't have to discard anything.

Q. What should an XPATH selector look like?

A. The following is an example of an XPATH selector. Pay careful attention to the use of double and single quotes.

```
String selector =
"JMS_BEA_SELECT('xpath', '/recipient/transport/text()') =
'email";
tsubscriber = tsession.createSubscriber(topic, selector, false);
```

JMS_BEA_SELECT is a built-in function in WebLogic JMS SQL syntax. You put it in your selector string when you create a consumer. Note the use of single quotes around xpath, the XML tab, and the string value.

Q. How do I handle request/response using WebLogic JMS?

A. There are several approaches to handling request/response processing with JMS.

- Use a temporary queue for each requestor and have the response go back to that queue.
- Use the QueueRequestor class, which does the temporary queue for you, and wait for the reply, as in the following:

```
// create temporary queue for receiving answer
qrequestor = new QueueRequestor(qsession, queue);
TextMessage msg = qsession.createTextMessage();
TextMessage reply = (TextMessage) qrequestor.request(msg);
```

- Use a dedicated response topic or queue with message selectors.

Q. How do I put a message back on the queue for processing?

A. The following are several approaches:

- Use a transacted session, then rollback the session so the message will go back to the queue.
- Use `Session.CLIENT_ACKNOWLEDGE` when creating a session, then recover the session so the message will go back to the queue.
- Use a JTA transaction, then rollback the transaction so the message will go back to the queue.

Q. Is it okay to add new sessions and subscribers to a Queue or Topic Connection once it has been started?

A. Yes, with one caveat. You may not add new subscribers/consumers to a session if it already has active async consumers. Sessions must only be accessed single-threaded

as per [Sun Microsystems' JMS specification](#) spec. If you feel you need to do this, create a new Session and add it to that one instead.

You can add receivers to a started connection. A receiver in itself is not asynchronous. You need a listener to make it asynchronous. The first creation of a receiver is always safe. If you then add a listener for that first receiver, you have to worry for any future receivers in that same session. You can create new sessions and the first receiver for that session with no worries.

Once you want to create a second receiver in a session, if the first receiver has a `MessageListener`, you have to take care to make sure there are no other threads of execution in that session. You can do this by stopping the connection or actually creating your receiver from the `onMessage` routine of the first receiver.

Q. What can I do when I get `java.lang.OutOfMemoryError` because producers are faster than consumers?

A. Quotas can be used to help this situation. Your sender will then receive `ResourceAllocationExceptions` and the server will stay up.

As of WLS 6.1 SP02 or later, you can use the Message Paging feature, which can free up valuable virtual memory during peak message load periods by swapping out messages from virtual memory to persistent storage when message loads reach a specified threshold. For more information, see “[Managing JMS](#)” in the *Administration Guide*.

Q. Why have different connection factories?

A. To get multiple different sets of connection attributes. Clients that need different behaviors should use different factories. If all of your clients need the same behavior, then one factory is sufficient.

Q. How should connections and sessions be allocated?

A. Think of a connection as a single physical connection (a TCP/IP link). A session is a means for producing and consuming an ordered set of messages. Creating a connection is generally expensive. Creating a session is less expensive. Generally people use one connection and share across all the threads with each thread having its own session. If you have thread groups and need to start/stop/close the resources for a given group, one connection per group is good. A group can have exactly one thread.

Q. How does an application know if an application server goes down?

A. There are two exception listeners that you can register. [Sun Microsystems' JMS specification](#) defines `Connection.setExceptionListener` that tells you if there is a problem with the connection. That means that all consumers under that connection are also in trouble. The reason you will get the connection exception is because the WebLogic server you connect to on the other side is dead or not responding or someone killed your connection via the Mbean interface.

However, for WebLogic Server JMS, you may have multiple sessions in a connection, with the sessions going to multiple backend servers. WebLogic Server has an extension for this called `WLSession.setExceptionListener` that tells you if there is a problem with a session. For more information see <http://e-doc.bea.com/wls/docs70/javadocs/weblogic/jms/extensions/WLSession.html>.

Q. Is there a way to dynamically change an existing selector for a `TopicConsumer` using `setMessageSelect(String s)`?

A. No. Once you instantiate the consumer the selector is fixed at the time that the consumer is created. Changing the selector is like removing the current consumer, removing all associated messages and then creating a new one.

Q. How can I avoid asynchronous message deadlocks?

A. Due to a limitation in [Sun Microsystems' JMS specification](#), asynchronous messages can become deadlocked if the `close()` method of a session is inside a user-synchronized block. To resolve this, you must move the `close()` method outside the user-synchronized block. For example:

```
public class CloseTest() {
private void xxx() {
    synchronized (this) {
        create connection/session/consumer
        initialize and set a listener for this consumer;
        wait();
        connection.close();
    }
}

private void onMessage(Message message) {
    synchronized (this) {
        notify();
    }
}
}
```

Before the `connection.close()` method is closed, another message can be delivered to the `onMessage` routine by the `JMSProvider`. The `main()` method thread owns the monitor lock for the `CloseTest` method. Before the `onMessage()` method of the `CloseTest` class fires, JMS sets `INLISTENER` as the state for the session in `JMSSESSION` (the [JMS specification](#) says that the `close()` method must wait for the `onMessage` routine), so that the `main()` method thread can wait for the `onMessage` routine to complete.

Now when the `onMessage` routine tries to acquire the monitor lock, it blocks waiting for the `main()` method thread to give up, and the `main()` method thread is waiting for the `onMessage` to be completed.

JMS also blocks when the `close()` method of a consumer is done from an `onMessage` routine and the `allowCloseInOnMessage` attribute is set to `false` in the `config.xml` file.

Q. What are the advantages of message-driven beans?

A. The message-driven bean is a stateless component that is invoked by the EJB container as a result of receiving messages from a JMS queue or topic. It then performs business logic based on the message contents, effectually freeing you from any JMS configuration and reconnection chores.

The message-driven bean model allows EJB developers to work with a familiar framework and set of tools, and also provides access to the additional support provided by the container. The goal of the message-driven bean model is to assure that developing an EJB that is asynchronously invoked to handle the processing of incoming JMS messages is as easy as developing the same functionality in any other `JMS MessageListener`.

One of the main advantages of using message-driven beans in place of the standard `JMS MessageListener` is that a JTA transaction can be started for you automatically and the received message will be part of that transaction. In this case, other operations can be infected with the same JTA transaction such as database operations. This is the only way to infect a message from an asynchronous consumer and another JTA operation with the same transaction.

For more information on message-driven beans, see [Designing Message-Driven Beans](#) in *Programming WebLogic Enterprise JavaBeans*.

Q. How does concurrency work for message-driven beans?

A. The way concurrency is achieved for Queues is by spawning one `JMSSession` per MDB instance in the pool. Since `JMSSessions` are processed in parallel by JMS, concurrency is obtained naturally this way and JMS takes care of delivering the message to, at most, one listener. If an MDB is deployed to multiple servers in a cluster, `JMSSessions` are created for each MDB instance on each server and load balancing will be done across them.

Within a single server, one topic consumer is used to pass out messages to multiple threads to get the concurrency while producing only a single copy of each message. You can configure multiple MDBs to listen on the same topic and each MDB will receive a copy of every message. When using multiple servers, each server gets its own consumer and therefore its own copy of each message. It is not currently possible to share a consumer across multiple servers. If you want a message to be processed by exactly one MDB, use a queue.

One customer had an example where topic MDBs are needed in which there will be multiple implementations of the MDBs listening on the same topic. In other words, more than one MDB with a different implementation may be subscribing to the same topic. The client has no advanced way of knowing how many different kinds of MDBs may be listening on the same topic, but it is possible for there to be more than one listener, therefore topics, not queues. For each kind of MDB listening on the topic, the message is delivered exactly once (i.e., the message will be delivered exactly once to an instance in each named MDB pool listening on the topic).

Q. Can an MDB be a message producer or both a producer and consumer?

A. Yes. You have no JMS context inside the MDB so you will need to establish a connection, session and producer yourself. One option is to do this every time you come into the `onMessage` routine for the MDB. This is sufficient if the message rate is relatively low. The second option is to establish the necessary objects in `ejbActivate()`. Note that the objects are not serializable so they can't be passivated for a stateful session bean or an entity bean. When the EJB deactivates, you need to close the associated objects. The third option is that you could build up a JMS connection/sender session pool within a startup class complete with your own synchronization and blocking to get a connection. There is an example in the answer to the question in this section, "Is it possible to send or receive a message from within a message listener?".

Q. If an MDB uses a durable subscription, will messages be accumulated if the MDB is not deployed?

A. Not at this time. Before the MDB is deployed the first time, no messages are accumulated for when it is deployed. This is different from passivated. Just because

the MDB is not currently active doesn't mean it isn't deployed. The container still maintains the subscription and gives out messages to the MDBs.

Q. How do I use non-WebLogic JMS provider destinations to drive MDBs?

A. See the “Using Foreign JMS Providers with WebLogic Server” white paper ([jmsproviders.pdf](#)) on <http://dev2dev.bea.com/technologies/jms/index.jsp> for a discussion on this topic.

Q. Can you use a foreign JMS provider to drive an MDB transactionally?

A. Yes. In WebLogic Server 7.0, you can deploy an MDB that supports container-managed transactions against a foreign JMS provider. If the MDB is configured with a “transaction-type” attribute of “Container” and a “trans-attribute” of “Required”, then WLS will use XA to automatically enlist the foreign JMS provider in a transaction. (See the next question for an example of an MDB that uses container-managed transactions.)

If the foreign JMS provider does not support XA, then you cannot deploy an MDB that supports container-managed transactions with that provider. Furthermore, if the JMS provider does support XA, you must ensure that the JMS connection factory that you specify in the `weblogic-ejb-jar.xml` file supports XA—each JMS provider has a different way to specify this.

See the white paper, “Using Foreign JMS Providers with WebLogic Server” ([jmsproviders.pdf.doc](#)) on <http://dev2dev.bea.com/technologies/jms/index.jsp> for an example of how to configure an MDB to use a foreign provider.

Q. How do I use JTA transactions within an MDB?

A. In the `ejb-jar.xml` file, define the transaction type as Container and the trans-attribute as Required, as in the following example:

```
<ejb-jar>
  <enterprise-beans>
    <message-driven>
      <ejb-name>MDB</ejb-name>
      <ejb-class>MDB</ejb-class>
      <transaction-type>Container</transaction-type>
      <message-driven-destination>
        <destination-type>javax.jms.Queue</destination-type>
      </message-driven-destination>
    </message-driven>
  </enterprise-beans>
</assembly-descriptor>
```

```

<container-transaction>
  <method>
    <ejb-name>MDB</ejb-name>
    <method-name>*</method-name>
  </method>
  <trans-attribute>
    Required
  </trans-attribute>
</container-transaction>
</assembly-descriptor>
</ejb-jar>

```

To roll back a transaction, you can either use the Weblogic extension `TXHelper` or you can use the MDB context as in the following code examples:

```

UserTransaction ut =
    weblogic.transaction.TXHelper.getUserTransaction();
ut.setRollbackOnly();

```

or

```

private MessageDrivenContext context;

public void setMessageDrivenContext(
    MessageDrivenContext mycontext) {
    context = mycontext;
}

public void onMessage(Message msg) {
try {    // some logic
}

    catch(Exception e) {
        System.out.println("MDB doing rollback");
        context.setRollbackOnly();
}
}

```

Q. How do server session pools and message driven beans compare?

A. MDBs listen on exactly one destination. Consumers listen on exactly one destination. `ConnectionConsumers` listen on exactly one destination.

A `ServerSessionPool` can have multiple `ConnectionConsumers`, which allows you to have a single `MessageListener` being fed by multiple consumers.

`ServerSessionPools` do not infect the `MessageListener` with a transaction, so the receipt of the message that initiated the listener is not part of the transaction. For an MDB, you can specify transaction as being `REQUIRED` and the driving message is part of the transaction.

`ServerSessionPools` have the limitation that the destination on which its connection consumers are listening must be hosted by the same JVM. That is, you cannot have a server session pool listening on a remote queue or topic and foreign (non-WebLogic Server JMS) destinations are not supported. MDBs can be distributed and can have foreign destinations.

MDBs currently receive one message per deployment (in Service Pack 1). If you have an MDB with multiple instances on a single server listening on a topic, it will receive only one copy of the published messages regardless of the number of instances. If you have an MDB deployed across multiple machines listening on a topic, each deployment will receive a copy of any published message on that topic. You will get multiple copies of the message distributed evenly across all of your MDB deployments.

1 FAQs: WebLogic Messaging Bridge

- Why did the messaging bridge fail to connect to the source bridge destination?
- I configured the messaging bridge to use the Exactly-once quality of service for two-phase transactions. So why am I getting a “quality of service is unreachable” error?
- Can I configure the messaging bridge to automatically downgrade the quality of service if the Exactly-once service isn’t available on either the source or target bridge destination?
- Why do I get a security authorization exception when attempting to forward messages from a WebLogic Server 7.0 GA, SP01, or SP02 destination to a release 6.1 destination?
- When configuring a source or target messaging bridge destination, do I need to set the Adapter Classpath field?
- Can the messaging bridge forward durable subscription messages between separate WebLogic Server 6.1 and release 7.0 or later domains?
- How do I enable debugging for the messaging bridge?
- What do the messaging bridge monitoring states indicate on the Monitor Messaging Bridge console page?
- Is there another way to monitor the messaging bridge without using the Administration Console?
- Can the messaging bridge use distributed destinations as source and target destinations?

Q. Why did the messaging bridge fail to connect to the source bridge destination?

A. Either an error occurred when configuring the source bridge destination parameters, or the actual source destination is not running and cannot communicate with the messaging bridge.

- Verify whether the bridge's source destination is correctly configured, by making sure that the following fields on the JMS Bridge Destination → Configuration → General console page have been properly completed:
 - Connection URL—this must be the URL of the JNDI provider used to look up the connection factory and actual destination.
 - Destination JNDI Name—this must be the JNDI name of the actual destination mapped to the source bridge destination.
 - Connection Factory JNDI Name—this must be the connection factory used to create a connection for the actual destination mapped to the source bridge destination.
 - User Name/Password—make sure that this user ID has permission to access the actual source destination.
- Verify that the actual source queue or topic destination mapped to the source bridge destination is running and healthy, as follows:
 - Is the WebLogic Server instance hosting the source destination running?
 - Is the JMS server hosting the source destination correctly deployed?

Note: This troubleshooting scenario for correcting a source bridge destination connection failure also applies to target bridge destinations.

Q. Can the messaging bridge handle two-phase or global transactions between separate WebLogic Server domains or between different releases?

A. Yes, as long as the communication is between source and target WebLogic domains that are both running release 6.1 SP03 or later, *and* the bridge is configured to use the Exactly-once quality of service.

Also, depending on which service pack of release 7.0 the bridge is running on, you may have to establish a trusted relationship between release 6.1 and 7.0, as follows:

- In order for a release 7.0 GA, SP01, and SP02 bridge to communicate with a release 6.1 domain, you must establish a trusted relationship across release 6.1 and 7.0 domains, as explained in “Why do I get a security authorization

exception when attempting to forward messages from a WebLogic Server 7.0 GA, SP01, or SP02 destination to a release 6.1 destination?” on page 1-52.

- For release 7.0 SP03 or later, you do not need to establish a trusted relationship across release 6.1 and 7.0 domains.

For more information about using the Exactly-once QOS when interoperating between release 6.1 and release 7.0 domains, see [“Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain”](#) in the *Administration Guide*.

Q. I configured the messaging bridge to use the Exactly-once quality of service for two-phase transactions. So why am I getting a “quality of service is unreachable” error?

A. There are some additional configuration requirements for the messaging bridge to handle transactions between WebLogic domains:

- The supported adapters are located in the `WL_HOME\server\lib` directory. For the Exactly-once QOS, the transaction adapter, `jms-xa-adp.rar`, must be deployed in the release 7.0 domain where the bridge is running, via the `select Deployments` → `Applications` node on the console.
- This `jms-xa-adp.rar` adapter must also be identified in the Adapter JNDI Name attribute as `eis.jms.WLSConnectionFactoryJNDIXA` on the JMS Bridge Destination → `Configuration` tab for both the source and target bridge destinations.
- For WebLogic JMS, verify that you are using the transactional `XAConnectionFactory` for the queue or topic destinations mapped to both the source and target bridge destinations. To verify this, the following attributes must be set on the JMS → `Connection Factory` → `Configuration` → `Transactions` console tab or in the configuration file (`config.xml`):

```
UserTransactionsEnabled=true  
XAConnectionFactory=true
```

- For third-party JMS vendors, verify that you are using a transactional connection factory for the destinations mapped to the source and target bridge destinations.

For more information about using the Exactly-once QOS when interoperating between release 6.1 and release 7.0 or later domains, see [“Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain”](#) in the *Administration Guide*.

Q. Can I configure the messaging bridge to automatically downgrade the quality of service if the Exactly-once service isn't available on either the source or target bridge destination?

A. Yes, just make sure to select the QOS Degradation Allowed check box on the Messaging Bridge → Configuration → General administration console page.

Q. Why do I get a security authorization exception when attempting to forward messages from a WebLogic Server 7.0 GA, SP01, or SP02 destination to a release 6.1 destination?

```
java.lang.SecurityException: Invalid Subject: principals=[user1]
```

A. In WebLogic Server 6.1, the trust relationship between two WebLogic Server domains was established if the system password was the same in both domains. For a release 7.0 GA, SP01, and SP02 messaging bridge to communicate with a release 6.1 domain, you must establish a trusted relationship across release the 6.1 and 7.0 domains. A trusted relationship is established when the Credential attribute for one domain matches the Credential attribute for another domain. Therefore, if you want a release 6.1 domain to interoperate with a release 7.0 SP02 or earlier domain, you need to change the Credential attribute in both domains to the password of the “system” user in the release 6.1 domain.

However, for release 7.0 SP03 or later, you do not need to establish a trusted relationship across release 6.1 and 7.0 domains.

- For more information about establishing a trust relationship between release 7.0 and 6.1 domains for the messaging bridge, see “[Enabling Security Interoperability for WebLogic Domains](#)” in the *Administration Guide*.
- For more information about release 6.1 domain interoperability security, see “[Using Compatibility Security](#)” in *Managing WebLogic Security*.
- For more information about WebLogic Server 7.0 domain interoperability security, see “[Enabling Trust Between WebLogic Domains](#)” in *Managing WebLogic Security*.

Q. I deployed the transactional `jms-xa-adp.rar` resource adapter on the WebLogic 7.0 domain where the message bridge is running, but I still get a “*failed to find bridge adapter*” message?

A. You need to associate *both* the source and target bridge destinations with the appropriate `.rar` adapters in order for the bridge to communicate with them. For the `jms-xa-adp.rar` transaction adapter, it must be identified in the Adapter JNDI Name

attribute as `eis.jms.WLSConnectionFactoryJNDIXA` on the JMS Bridge Destination →Configuration tab for both the source and target bridge destinations.

Note: The “failed to find bridge adapter” message does not necessarily indicate a problem if it only occurs once. However, if it occurs repeatedly, you should check the adapter deployment and the adapter JNDI name used in the source and target bridge destinations.

For more information about the bridge resource adapters, see “[About the Bridge’s Resource Adapters](#)” in the *Administration Guide*.

Q. When configuring a source or target messaging bridge destination, do I need to set the Adapter Classpath field?

A. Leave the Adapter Classpath field blank when connecting to source and target destinations that are both running on release 7.0 or later. When connecting to either a source or target destination that is running on release 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`.

Q. Can the messaging bridge forward durable subscription messages between separate WebLogic Server 6.1 and release 7.0 or later domains?

A. Yes, as long as the domain hosting the bridge is using WebLogic 7.0 Service Pack 1 or later. To enable durable messages across the messaging bridge using the Administration Console, select the Durability Enabled attribute on the Messaging Bridge →Configuration →General tab.

Q. How do I enable debugging for the messaging bridge?

A. You can enable debugging for the messaging bridge using either of the followings methods:

- Add the following lines to your WebLogic start script (before the `weblogic.Server` line):

```
-Dweblogic.Debug.DebugMessagingBridgeStartup=true
```

```
-Dweblogic.Debug.DebugMessagingBridgeRuntime=true
```

- Add the following statements to the `ServerDebug` entry in your configuration file (`config.xml`) for the server that the messaging bridge is running on:

```
DebugMessagingBridgeStartup="true"
```

```
DebugMessagingBridgeRuntime="true"
```

Once debugging is enabled for the messaging bridge, the debugging messages are sent to the server log by default. However, if you want them to appear in the Administration Console, add “DumpToConsole” to the statements show above. For example:

```
-Dweblogic.Debug.DebugMessagingBridgeStartupDumpToConsole=true
```

Q. What do the messaging bridge monitoring states indicate on the Monitor Messaging Bridge console page?

A. When monitoring a messaging bridge’s state on the Server →Services →Monitor →Messaging Bridge page, use the following table to determine a course of action, if necessary.

Description	Action
WARN: Failed to find the source adapter	Check if the adapter is deployed or the JNDI name in the source JMSBridgeDestination instance is correct.
WARN: Failed to find the target adapter	Check if the adapter is deployed or the JNDI name in the target JMSBridgeDestination instance is correct.
Found both of the adapters and making connections	No.
WARN: Stopped by the administrator	No.
WARN: Failed to look up the source adapter	Check if the adapter is deployed or the JNDI name in the source JMSBridgeDestination instance is correct.
WARN: Failed to look up the target adapter	Check if the adapter is deployed or the JNDI name in the target JMSBridgeDestination instance is correct.
Found two adapters and about to make connections	No.
WARN: Failed to connect to the source	Check all the parameters configured for the source bridge destination. Check if the source server is running and whether the actual destination is active.
Connected to the source	No.

Description	Action
WARN: Failed to connect to the target	<p>Check all the parameters configured for the target bridge destination.</p> <p>Check if the target server is running and whether the actual destination is active.</p>
Connected to the target	No.
Forwarding messages	No.
WARN: Failed to connect and will reconnect later	Check if the source and target bridge destinations are running and healthy.

Q. Is there another way to monitor the messaging bridge without using the Administration Console?

A. Yes, there is a run-time MBean (`MessagingBridgeRuntimeMBean`) for each bridge instance. WebLogic Server run-time MBeans provide a snapshot of information about domain resources. When a particular resource in the domain (such as a messaging bridge) is instantiated, an MBean instance is created which collects information about that resource.

The `MessagingBridgeRuntimeMBean` has a `getState()` method that currently returns a String (“Active” or “Inactive”) and a `getDescription()` method, which returns a String with more detailed information. The name of a bridge runtime MBean consists of the WebLogic Server instance name and the bridge name. If a bridge named *mybridge*, runs on WebLogic Server instance named *myserver*, the bridge runtime MBean will be named *myserver.bridge.mybridge*.

For more information about using run-time Mbean management commands, see “[WebLogic Server Command-Line Interface Reference](#)” in the *Administration Guide*. For more information about programming MBean monitoring notifications, see “[Using WebLogic Server MBean Notifications and Monitors](#)” in *Programming WebLogic JMX Services*.

Q. Can the messaging bridge use distributed destinations as source and target destinations?

A. Yes, the 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 destinations 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.

2 FAQs: JTA

- Can I use a non-XA driver in distributed transactions?
- Can I use more than one non-XA connection pool in distributed transactions?
- How do XA and non-XA drivers differ in distributed transactions?
- What XA drivers can I use in addition to the WebLogic jDriver for Oracle/XA?
- Can I use the Oracle thin driver as an XA driver in distributed transactions?
- Why do I get SQLException “Result set already closed” message?
- Do I need a 2PC licence when I use JMS with one JDBC non-XA driver?
- Why am I getting an exception when I use JMS with a non-XA driver?
- Can I obtain a JDBC connection before I start a distributed transaction?
- Can I close a JDBC connection after the distributed transaction is committed or rolled back?
- I get the following XAER_RMFAIL XAException when accessing an XAResource: "Internal error: XAResource '<name>' is unavailable". What does that mean? How should I handle it?

Q. How can I incorporate MQSeries as an XA resource for distributed transactions in WebLogic Server?

A. You can download a zip file with instructions, support classes, utilities, and an example from the [code samples for weblogic server](#) page on BEA’s *dev2dev* site.

Q. Can I use a non-XA driver in distributed transactions?

A. When the non-XA connection pool is the only resource participating in a transaction distributed across multiple servers, you just need to configure a `TxDataSource` for the non-XA driver.

However, when more than one resource participates in the distributed transaction, you must also set the `TxDataSource` property `EnableTwoPhaseCommit=true`. For more information, see [Managing JDBC Connectivity](#) in the *Administration Guide*. In both cases, always obtain a connection via the `DataSource` interface, not through the deprecated `DriverManager` interface. If you obtain a connection via `DriverManager`, the interface cannot pick up the `EnableTwoPhaseCommit` setting of the `TxDataSource`; this may result in unexpected behavior in distributed transactions. Also, when you use the `DataSource` interface, you do not need to distinguish either the URL or the specific WebLogic multitier driver (JTS, RMI, or pool.) The URL and specific driver are obtained through the `config.xml` file and JNDI lookup.

Q. Can I use more than one non-XA connection pool in distributed transactions?

A. No. Even if you set `EnableTwoPhaseCommit=true` for both `TxDataSources` of the connection pools, attempting to use two non-XA connection pools in the same distributed transaction will result in:

```
"java.sql.SQLException: Connection has already been created in this tx context for pool named <first pool's name>. Illegal attempt to create connection from another pool: <second pool's name>"
```

when you attempt to get the connection from the second non-XA connection pool.

Q. How do XA and non-XA drivers differ in distributed transactions?

A. The differences between XA and non-XA JDBC drivers are:

- **Atomicity Guarantee.** An XA driver implements the `XAResource` interface and can participate fully in the 2PC protocol driven by the WLS Transaction Manager. This guarantees atomicity of updates across multiple participating resources.

However, a non-XA driver does not implement the `XAResource` interface and cannot fully participate in the 2PC protocol. When using a non-XA driver in a distributed transaction, WLS implements the `XAResource` wrapper on behalf of the non-XA driver. If the data source property `enableTwoPhaseCommit` is set to `true`, then the WLS `XAResource` wrapper returns `XA_OK` when the Transaction Manager invokes the `prepare()` method. When the Transaction Manager invokes `commit()` or `rollback()` during the second phase, the WLS

XAResource wrapper delegates the `commit()` or `rollback()` call to the non-XA JDBC connection. Any failure during `commit()` or `rollback()` results in heuristic exceptions. Application data may be left in an inconsistent state as a result of heuristic failure. If one is using transactions across multiple resource managers, an XA-Compliant driver is required in order to guarantee atomicity.

- **Redirecting Connections.** A non-XA driver can be configured to perform updates in the same distributed transaction from more than one process, as explained in [Can I use a non-XA driver in distributed transactions?](#). WLS internally redirects the JDBC calls made from different processes to the same physical JDBC connection in one process. However, when you use a XA driver, no such redirection will be done. Each process will use its own local XA database connection, and the database ensures that all the distributed updates made in the same distributed transaction from different processes will be committed atomically.
- **Connection Management.** Whether you are using the non-XA driver or XA driver in distributed transactions, WLS implements JDBC wrappers that intercept all the JDBC calls and obtains a physical JDBC connection from the connection pool on demand.
 - When you use a non-XA driver in distributed transactions, in order to ensure that updates made from different processes are committed atomically, WLS associates the same physical JDBC connection with the distributed transaction until it is committed or rolled back. As a result, the number of active distributed transactions using the non-XA connection pool is limited by the maximum capacity of the JDBC connection pool.
 - When you use an XA driver, the connection management is more scalable. WLS does not hold on to the same physical XA connection until the transaction is committed or rolled back. In fact, in most cases, the XA connection is only held for the duration of a method invocation. WLS JDBC wrappers intercept all JDBC calls and enlist the XAResource associated with the XA connection on demand. When the method invocation returns to the caller, or when it makes another call to another server, WLS delists the XAResource associated with the XA connection.
 - WLS also returns the XA connection to the connection pool on delistment if there are no open result sets. Also, during commit processing, any XAResource object can be used to commit any number of distributed transactions in parallel. As a result, neither the number of active distributed transactions using the XA connection pool nor the number of concurrent commit/rollbacks is limited by the maximum capacity of the connection pool.

Only the number of concurrent database access connections is limited by the maximum capacity of the connection pool.

Q. What XA drivers can I use in addition to the WebLogic jDriver for Oracle/XA?

A. Theoretically, you can use any third party XA driver that is compliant with the JDBC 2.0 standard extension specification with WLS. However, an individual vendor's XA driver may have bugs that prevent it from working properly.

Refer to JDBC Configuration guidelines for details about how to configure them at <http://e-docs.bea.com/wls/docs70/adminguide/managetx.html>.

Q. Can I use the Oracle thin driver as an XA driver in distributed transactions?

A. Oracle 8.1.7 thin driver has threading problems, so BEA developed the following workaround: We use a dedicated XA connection for the duration of prepare, commit, and rollback operation. This is different from the default XA connection management model in that any XAResource object is used to commit any number of transactions in parallel. This limits the number of concurrent commits to the max capacity of the XA connection pool. Note that this workaround is an Oracle specific workaround and will not affect the usage of other XA drivers.

Q. Why do I get SQLException "Result set already closed" message?

Problem: I am using WebLogic jDriver for Oracle/XA (transaction mode) from the client side. Updating in a distributed transaction works fine. However, when I try to perform a query, I get `SQLException Result set already closed`. How do I work around this?

A. WebLogic jDriver for Oracle has a limitation that closes all open result sets when the method returns to the caller.

Using the driver from the server side, for example, in a bean, does not have this limitation. Using the driver from the server side is also recommended from application architecture and performance perspective. Using the driver from the client side incurs round-trip cost for every JDBC call being made.

This limitation exists because WebLogic jDriver for Oracle XA is implemented using Oracle's OCI API and C XA switch, and there is an Oracle problem when using OCI with XA in multi-threaded mode. Closing an OCI cursor in a thread that is different than the thread in which it is opened may result in server crash or unexpected behavior. As a result, the WebLogic driver implicitly closes all open result sets upon returning a call to the caller.

Q. Do I need a 2PC licence when I use JMS with one JDBC non-XA driver?

A. Yes, you do. JMS is also a XAResource that participates in the distributed transaction. Therefore, there are two resources participating in the distributed transaction, and a 2PC license is needed.

Q. Why am I getting an exception when I use JMS with a non-XA driver?

Problem: I am using JMS with one JDBC non-XA driver. Transaction fails to commit with the following exception `javax.transaction.xa.XAException: JDBC driver does not support XA, hence cannot be a participant in two-phase commit.`

A. As mentioned in the previous question [“Do I need a 2PC licence when I use JMS with one JDBC non-XA driver?”](#), JMS is also a XAResource that participates in the distributed transaction. When more than one resource is participating in the distributed transaction, you need to set the data source property `EnableTwoPhaseCommit=true` as explained in [“Can I use a non-XA driver in distributed transactions?”](#)

Q. Can I obtain a JDBC connection before I start a distributed transaction?

A. This depends on whether you are using a non-XA or XA driver.

- When you use a non-XA driver in a distributed transaction, always obtain a JDBC connection after the distributed transaction is begun.
- If you are using an XA driver, you can obtain the connection before or after the distributed transaction begins.

Q. Can I close a JDBC connection after the distributed transaction is committed or rolled back?

A. For both non-XA and XA driver, you can close the connection after the distributed transaction is completed.

Q. I get the following XAER_RMFAIL XAException when accessing an XAResource: "Internal error: XAResource '<name>' is unavailable". What does that mean? How should I handle it?

A. JTA has its own resource health monitoring that works as follows:

A resource is considered active either if there are no pending requests or if we get a result from any of the `XAResource` pending requests that is not an `XAER_RMFAIL`. If an `XAResource` is not active within the two minutes, it is declared dead. Any further requests to the `XAResource` are shunned, and an `XAER_RMFAIL` `XAException` as above is thrown. The intent is to prevent further loss of threads if the `RM` is dead.

A resource is declared active again, if you re-register the `XAResource` with the WebLogic Server Transaction Manager by calling `weblogic.transaction.TransactionManager.unregisterResource` followed by `registerStaticResource` or `registerDynamicResource`, or after a timeout period of 30 minutes. If you are using WLS JDBC connection pools, you only need to enable the JDBC connection pool refresh feature (by specifying the "RefreshMinutes" property of the connection pool), and, upon a successful connection pool refresh, the corresponding `XAResource` will be re-registered automatically. If you are registering your own `XAResource`, either via `weblogic.transaction.TransactionManager.registerStaticResource` or `registerDynamicResource` APIs, you will need to re-register the `XAResource` by calling `weblogic.transaction.TransactionManager.unregisterResource` followed by `registerStaticResource` or `registerDynamicResource`.

In general, a good way to debug potential `RM` problems is to turn on JTA XA debugging, by specifying `-Dweblogic.Debug=weblogic.JTAXA` as JVM parameter on WLS startup.

3 FAQs: Plug-Ins

- How does the Apache HTTP Server Plug-In work?
- How does the plug-in route the request for a sticky session?
- What is new for debugging a plug-in in WebLogic Server 6.0?
- What is expected in the wproxy.log?
- What has changed in the 6.1 plug-in?
- What is static list, dynamic list, and general list?
- Does the 6.1 plug-in support two-way SSL?
- Sometimes a response from WebLogic Server to the plug-in contains the Set-Cookie header. Is this normal?
- If I install mod_wl_ssl.so with mod_perl to Apache 1.3.19, why does Segmentation Fault (11) occur in mod_wl_ssl.so when I access WebLogic via the plug-in?
- How do I install Apache 2.0.x so that I do not get the "Can't dlopen() a library containing Thread Local Storage: /usr/lib/libcl.2" error?

Q. How does the Apache HTTP Server Plug-In work?

A. For information on how a plug-in works, see [Installing and Configuring the Apache HTTP Server Plug-In](#) in *Using Web Server Plugins With WebLogic Server*.

Q. How does the plug-in route the request for a sticky session?

A. If the browser sends a cookie, we look for "JSESSIONID" (configurable by a parameter called "CookieName") in the "Cookie: " header.

If the cookie is disabled and URL re-writing is used, the session id is encoded in the URL. In WebLogic Server 5.1 and earlier, it was encoded in the query string:

```
?WebLogicSession=my_dumy_session
```

In WebLogic Server 6.0 and later, it was encoded in the parameter:

```
;JSESSIONID=my_dummy_session
```

If no session is found in the query string or parameter and if it is small enough to be read into memory, WebLogic Server looks for the session in the postdata.

Q. What is new for debugging a plug-in in WebLogic Server 6.0?

```
"Debug = ON" logs only informational and error messages
HFC : headers from the client, informational, and error messages
HTW : headers sent to wls, informational and error messages
HFW : headers sent from wls, informational and error messages
HTC : headers sent to the client, informational and error messages
ALL : everything
OFF : nothing -- default(should be used in production)
```

The log file is configurable for 6.1. For later versions of WebLogic Server, the `WLLogFile` was introduced to configure the debug filename and location.

Q. What is expected in the `wlproxy.log`?

Every request looks like the following:

```
"=====New Request: [GET / HTTP/1.1] ====="
PathTrim, DefaultFileName, and PathPrepend will be performed in
order. The final request will be logged as the following: "Fri Jun
22 14:24:40 2001 The request string is '/index.jsp'"
```

Looking for session information and determining the primary:

```
"Fri Jun 22 14:24:40 2001 Initializing lastIndex=0 for a list of
length=1 Fri Jun 22 14:24:40 2001 create a new server node:
id='qa89:443' server_name='mint.beasys.com', port='18080'"
```

Init SSL if SecureProxy is set to ON:

```
"Fri Jun 22 14:24:40 2001 INFO: SSL is configured Fri Jun 22
14:24:40 2001 INFO: Initializing SSL library Fri Jun 22 14:24:40
2001 Loaded 1 trusted CA's Fri Jun 22 14:24:40 2001 INFO:"
```

Successfully initialized SSL Fri Jun 22 14:24:40 2001 INFO: SSL configured successfully"

Initial connection being made:

```
"Fri Jun 22 14:24:40 2001 general list: trying connect to
'172.17.9.180'/443 Fri Jun 22 14:24:40 2001 Connected to
172.17.9.180:443"
```

Client headers and the post data (if present) being read:

```
"Fri Jun 22 14:24:40 2001 Hdrs from clnt:[Accept]=[image/gif,
image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/msword,
application/vnd.ms-powerpoint, */*]"
```

Client headers and the post data (if any) being sent:

```
"Fri Jun 22 14:24:40 2001 Hdrs to WLS:[Accept]=[image/gif,
image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/msword,
application/vnd.ms-powerpoint, */*]"
```

Response headers being obtained from WebLogic Server:

```
"Fri Jun 22 14:24:46 2001 Hdrs from
WLS:[Set-Cookie]=[JSESSIONID=OzI19WqYmFnRviHEu5gKlvot42ABeD8NPWnF
0jW6cawSGcrp2mru!4038528127411848936!-1408169548!80!443; path=/]
Fri Jun 22 14:24:46 2001 parsed all headers OK"
```

Response headers being sent to WebLogic Server and indicating if the connection is close or keep-alive is here:

```
"Fri Jun 22 14:24:46 2001 Hdrs to client:[Date]=[Fri, 22 Jun 2001
21:24:48 GMT] Fri Jun 22 14:24:46 2001 Hdrs to
client:[Server]=[WebLogic WebLogic Server 6.1 beta 06/21/2001
10:44:44 #122398 - internal build by jlee on client jlee.qa89] Fri
Jun 22 14:24:46 2001 canRecycle: conn=1 status=200 isKA=0 clen=2705
isCTE=0 Fri Jun 22 14:24:46 2001 closeConnection in load_utils:
deleting URL* Fri Jun 22 14:24:46 2001 INFO: Closing SSL context
Fri Jun 22 14:24:46 2001 INFO: sysSend 23 Fri Jun 22 14:24:46 2001
INFO: Error after SSLclose, socket may already have been closed by
peer Fri Jun 22 14:24:46 2001 r->status=200 returning 0"
```

Q. What has changed in the 6.1 plug-in?

A. The following has changed:

- HTTP1.1 support -- chunk-transferred and keep-alive (except for apache1.3.x)

- Session parsing (this breaks the backward compatibility)
- SSL support from the plug-in to WebLogic Server

Q. What is static list, dynamic list, and general list?

A. They are defined as follows:

- Static list: the initial server list defined in the config file
- Dynamic list: the current server list sent by wls upon a successful request
- General list: the current server list (could be static or dynamic) besides the primary and secondary servers associated with the current request

Q. Does the 6.1 plug-in support two-way SSL?

A. No. But the plug-in can be set-up to require the client certificate and pass it on to WebLogic Server. For example:

```
apache ssl

SSLVerifyClient require

SSLVerifyDepth 10

SSLOptions +FakeBasicAuth +ExportCertData +CompatEnvVars
+StrictRequire
```

Q. Sometimes a response from WebLogic Server to the plug-in contains the Set-Cookie header. Is this normal?

A. Yes. WebLogic Server will send the Set-Cookie header in the response if the Cookie header is missing from the request or X-WebLogic-Force-Cookie is detected. If the plug-in fails to connect to a bad server, the plug-in will send the header X-WebLogic-Force-Cookie: true to the next available server to force the client to update the corresponding cookie which contains the correct session information.

Q. If I install mod_wl_ssl.so with mod_perl to Apache 1.3.19, why does Segmentation Fault (11) occur in mod_wl_ssl.so when I access WebLogic via the plug-in?

A. The server could be any 6.x and the operating system could be any version of Solaris. The environment is WebLogic Server 6.x (Use mod_wl_ssl.so), Solaris 2.x.

In order to avoid getting Segmentation Fault (11), add a VirtualHost block for HTTP as in the following example:

```
# General setup for the virtual host
DocumentRoot "/export/home/happy/local/apache_1.3.19/htdocs"
ServerName happy1
ServerAdmin happy@happyville
ErrorLog /export/home/happy/local/apache_1.3.19/logs/error_log
TransferLog /export/home/happy/local/apache_1.3.19/logs/access_log
</VirtualHost>
```

For the SSL port, also use the IP address in:

```
<VirtualHost 206.189.223.111:443>
```

Use any valid DNS name for the ServerName.

Q. How do I install Apache 2.0.x so that I do not get the "Can't dlopen() a library containing Thread Local Storage: /usr/lib/libcl.2" error?

A. To install Apache 2.0.x so that you do not get this error, set the following environment variable:

1) before building the Apache server export CFLAGS="-lcl -lpthread"

2) before starting the Apache server export LD_PRELOAD="/usr/lib/libcl.2"

Follow these steps to build the Apache2.0.x server:

1)export CFLAGS="-lcl -lpthread"

2)./configure --prefix=\$INSTALLATION_DIRECTORY --enable-so
--with-mpm=worker

3)make

4)make install

4 FAQs: Server-Related Questions

- What should I do if my server “hangs” or “freezes”?
- How do I configure WebLogic to use a SOCKS proxy?
- How can I speed up connection responses?
- How does WebLogic support CORBA and client communication via IIOP?
- How do I speed up HTTP tunneling?
- Can WebLogic Server start with a UNIX boot?
- What affects servlet performance besides clients and traffic?
- Why do I get “NoClassDefFound”/“Too Many Open files” messages on Solaris?
- How do I increase WebLogic Server memory?
- Java-CORBA integration: RMI-IIOP or Java IDL?
- How do an RMI-IIOP application and an existing CORBA object interoperate?
- What is the function of T3 in WebLogic Server?
- How can I debug the Java code that I have running in WebLogic Server?

Q. What should I do if my server “hangs” or “freezes”?

A. If your WebLogic Server “freezes,” you will need to collect certain diagnostic information, including thread dumps and Java garbage collection metrics, before contacting BEA Technical Support. For details, see [Generate a Log File](#) and [Check Garbage Collection](#).

Q. How do I configure WebLogic to use a SOCKS proxy?

A. You can configure a java.net socket to use SOCKS by setting a Java system property. For details, see [How do I make Java work with a proxy server?](http://www.sunsite.unc.edu/javafaq/javafaq.html#proxy) at <http://www.sunsite.unc.edu/javafaq/javafaq.html#proxy>. Once the property is set, WebLogic socket connections use the SOCKS proxy.

Q. How can I speed up connection responses?

A. Connection delays are often caused by DNS problems. WebLogic performs a reverse lookup on the hostname from which a new connection is made. If the DNS reverse lookup is not working properly because the connection is coming from a proxy server, it could be responsible for the delay. You might want to work with your system administrator to determine whether DNS and the third-party networking software are working properly. Try writing a simple server program that performs a reverse lookup on any connection made to it. If that lookup is delayed, then you know that the proxy server is the source of the problem.

Q. How does WebLogic support CORBA and client communication via IIOP?

A. “CORBA” support means many things to many people. It often means simply IIOP and ORB support and does not use many CORBA services. WebLogic supports CORBA in multiple ways. The WebLogic Server 7.0 implementation of RMI-IIOP allows you to: connect Java RMI clients to WebLogic Server using the standardized IIOP protocol; connect CORBA/IDL clients, including those written in C++, to WebLogic Server; interoperate between WebLogic Server and Tuxedo clients; connect a variety of clients to EJBs hosted on WebLogic Server. For more information, see [Using WebLogic RMI over IIOP](http://e-docs.bea.com/wls/docs70/rmi_iiop/index.html) at http://e-docs.bea.com/wls/docs70/rmi_iiop/index.html.

Q. How do I speed up HTTP tunneling?

A. Unfortunately, a significant performance hit occurs when you use HTTP tunneling. We have optimized it somewhat, but, because everything is encapsulated in HTTP, HTTP tunneling is slower than direct Java-to-Java TCP/IP connections.

Be sure that you really need to use HTTP tunneling. For example, if your firewall can pass IP packets through port 80, you can use the fast T3 protocol on port 80.

If you must use HTTP tunneling to go through a firewall, e-Border has a product that performs better than HTTP proxies.

Q. Can WebLogic Server start with a UNIX boot?

A. You can add a startup script to your UNIX rc scripts to run WebLogic Server at UNIX boot time. For information about creating startup scripts, refer to [Starting an Administration Server Using a Script](#) in the *WebLogic Server Administration Guide*.

Q. What affects servlet performance besides clients and traffic?

A. Response time for a servlet is about 5 times slower when you are running a screen saver on the machine, particularly for the OpenGL screen savers. Try turning off your screen saver and see if that helps!

Q. Why do I get “NoClassDefFound”/“Too Many Open files” messages on Solaris?

Problem: When I am using WebLogic Server on Solaris and try to run my application, I get a “NoClassDefFound” error, although the class causing the error does exist and is in the right directory. In fact, there are other classes in the same directory that are getting loaded. I also get a “Too many open files” error.

A. We have seen this situation when the user account runs out of file descriptors. On Solaris, each user account has a certain limited number of file descriptors. You can find out how many file descriptors you have with the `limit` command in `csh`.

You can increase file descriptors if you have enough privileges with the `ulimit` command in the `csh`. Otherwise, ask your system administrator to increase the file descriptors available to your processes.

Q. How do I increase WebLogic Server memory?

A. Increase the allocation of Java heap memory for WebLogic Server. (Set the minimum and the maximum to the same size.) This example starts the server with a fixed heap size of 200MB:

```
$ java ... -ms200m -mx200m ...
```

This allocates 32 megabytes of Java heap memory to WebLogic Server, which improves performance and allows WebLogic Server to handle more simultaneous connections. You can increase this value if necessary.

Q. Java-CORBA integration: RMI-IIOP or Java IDL?

A. It is important to understand the distinction between these two ways of integrating Java with CORBA.

RMI-IIOP is for Java programmers who want to program to the RMI interfaces but use IIOP as the underlying transport. RMI-IIOP provides interoperability with other CORBA objects implemented in various languages, but only if all the remote interfaces are originally defined as Java RMI interfaces. It is of particular interest to programmers using Enterprise JavaBeans (EJBs), because the remote object model for EJB is RMI-based. It also allows you to use the standardized IIOP protocol and have a lightweight (minimal weblogic classes) client.

Java IDL is for CORBA programmers who want to program in Java based on interfaces defined in CORBA IDL. This is "business as usual" CORBA programming, supporting Java in exactly the same way as other languages like C++ or COBOL. If you want to integrate C++ (or any other language your orb supports into IDL) you would use the CORBA IDL method of programming.

Q. How do an RMI-IIOP application and an existing CORBA object interoperate?

A. If the existing CORBA object has its remote interfaces defined originally in CORBA IDL, then interoperability is not possible. RMI-IIOP applications can interoperate with other CORBA objects only when their remote interfaces are originally defined as Java RMI interfaces.

For example, to interoperate between an RMI-IIOP client and a C++ object you need to:

1. Define the remote interface of the object in Java as an RMI interface..
2. Run `rmic -idl` against the interface to produce IDL compatible with the RMI interface.
3. Run a C++ stub compiler against the IDL file to produce the C++ skeleton for your C++ server object.

Q. What is the function of T3 in WebLogic Server?

A. T3 provides a framework for WebLogic Server messages that support for enhancements. These enhancements include abbreviations and features, such as object replacement, that work in the context of WebLogic Server clusters and HTTP and other product tunneling.

T3 predates Java Object Serialization and RMI, while closely tracking and leveraging these specifications. T3 is a superset of Java Object. Serialization or RMI; anything you can do in Java Object Serialization and RMI can be done over T3.

T3 is mandated between WebLogic Servers and between programmatic clients and a WebLogic Server cluster. HTTP and IIOP are optional protocols that can be used to communicate between other processes and WebLogic Server. It depends on what you want to do. For example, when you want to communicate between

- A browser and WebLogic Server-use HTTP
- An ORB and WebLogic Server-IIOP.

Q. How can I debug the Java code that I have running in WebLogic Server?

A. You can use tools such as WebGain, JBuilder, NetBeans and JDB that rely on the Java Platform Debugger Architecture (JPDA) to debug your Java code running in WebLogic Server.

JPDA is integrated in the Java 2 Platform, Standard Edition (J2SE) SDK 1.3 on all platforms and SDK 1.2.2 for Linux. There is a [download](#) available from Sun to add JPDA support to the J2SE SDK 1.2.2 on Solaris and Microsoft Window platforms. If you are using J2SE SDK 1.2.2 on these platforms you must first get this download.

To allow a debugger to attach to the virtual machine that WebLogic runs you have to start WebLogic in debug mode. In order to start WebLogic in debug mode using a Sun virtual machine follow these steps (start with step one only if using a Solaris platform):

1. If using a Solaris platform, change the `LD_LIBRARY_PATH` environment variable to prepend `$JAVA_HOME/lib/sparc`:

```
export LD_LIBRARY_PATH=$JAVA_HOME/lib/sparc:$LD_LIBRARY_PATH
```

2. Add the following parameters to the `java` command line (before the "weblogic.Server" string) that launches WebLogic server:

```
-Xdebug  
-Xnoagent  
-Xrunjdpw:transport=dt_socket  
server=y  
address=<port_for_debugger_to_connect>  
suspend=n  
-Djava.compiler=NONE
```

Note that with the Hotspot Performance engine the `-Xnoagent` and `-Djava.compiler=NONE` options are no longer required, but are accepted and ignored for compatibility reasons.

If `server=y` and no `address` parameter is supplied, WebLogic Server chooses the transport address and prints it to the standard output stream. So, if a line such as:

4 *FAQs: Server-Related Questions*

```
Listening for transport dt_socket at address: 46666
```

prints in your standard output stream when the server starts, the number 46666 is the port number to be supplied to your tool's remote debugger in order to attach it to WebLogic's virtual machine.

5 FAQs: Server-Side Java (Servlets)

- How do I call a servlet with parameters in the URL?
- How can I run multiple instances of the same servlet class in the same WebLogic Server instance?
- How do I deserialize an `httpSession`?

Q. How do I call a servlet with parameters in the URL?

A. The usual format of a servlet parameter is a *name=value* pair that comes after a question-mark (?) at the end of the URL. To access these parameters, call the `getParameter()` method on the `HttpServletRequest` object, then write code to test the strings. For example, if your URL parameters are “`func=topic,`” where your URL appears as:

```
http://www.myserver.com/myservlet?func=topic
```

then you could parse the parameter as follows, where “`req`” is the `HttpServletRequest` object:

```
String func = req.getParameter("func");
if (func.equalsIgnoreCase("topic")) {
    . . . do some work
}
```

Q. How can I run multiple instances of the same servlet class in the same WebLogic Server instance?

A. If you want to run multiple instances, your servlet will have to implement the `SingleThreadModel` interface. An instance of a class that implements the `SingleThreadModel` interface is guaranteed not to be invoked by multiple threads

simultaneously. Multiple instances of a `SingleThreadModel` interface are used to service simultaneous requests, each running in a single thread.

When designing your servlet, consider how you use shared resources outside of the servlet class such as file and database access. Because there are multiple instances of servlets that are identical, and may use exactly the same resources, there are still synchronization and sharing issues that must be resolved, even if you do implement the `SingleThreadModel` interface.

Q. How do I deserialize an `HttpSession`?

A. To deserialize an `HttpSession`, construct a utility class that uses the current thread's `ContextClassLoader` to load the user defined objects within the application context. then add this utility class to the system `CLASSPATH`.

6 FAQs: Security

- What is the difference between the Compatibility realm and myrealm? Under what circumstances should I use each of these realms?
- What are the default groups users and everyone used for?
- Is there still a guest user?
- I want to provide additional fields in my Web application for form-based authentication. What application programming interfaces (APIs) should I use?
- I am using the 6.x security realm APIs in my application. How do I upgrade this functionality to the security APIs in WebLogic Server 7.0?
- Does WebLogic Server support Diffie-Hellman or DSS/DSA digital certificates?
- Can a Weblogic Server deployment have one RSA and one non-RSA certificate?
- Must we pay RSA licensing costs for non-RSA client code?
- How do I use Netscape security certificates with WebLogic Server?
- How do I restrict access to servlets and JSPs?
- Can I use RSA encryption algorithms and javax.crypto.* API to build applications?
- Can I use a JNDI Initial Context to pass the security credentials of a WebLogic Server user?
- Are WebLogic Server passwords secure?
- Why do I get a Java security permission error when starting and running WebLogic Server?
- Why do I get a certificate configuration error when I start the my Weblogic Server?

- Why can't I establish an outbound SSL connection when using the demonstration certificates?
- Why do I get a configuration error when establishing an SSL connection to WebLogic Server?
- Why doesn't my Custom security provider show up (that is, it doesn't appear as a `Configure a new Security Provider Type` link) in the Administration Console?
- Why do I get an exception when configuring a custom security provider through the WebLogic Server Administration Console?
- Why can't the Windows NT security realm in the CompatibilityRealm authenticate my Windows NT user?

Q. What is the difference between the Compatibility realm and myrealm? Under what circumstances should I use each of these realms?

A. If you have a `6.x config.xml` file and you boot WebLogic Server 7.0, the following realms are created:

- Compatibility realm—Allows you to use an existing 6.x security configuration as is in the management environment provided in WebLogic Server 7.0. The Realm Adapter providers allows access to existing stores of users, groups, and access control lists (ACLs).
- myrealm—Is the default security realm in WebLogic Server 7.0. By default, the WebLogic Security providers are configured in myrealm.

For more information, see *Managing WebLogic Security*.

Q. What are the default groups users and everyone used for?

A. The `users` and `everyone` groups are convenience groups that allow you to apply global roles and security policies. All WebLogic Server users are members of the `everyone` group. Only WebLogic Servers who are not the `<anonymous>` user are members of the `users` group.

For more information, see *Managing WebLogic Security*.

Q. Is there still a guest user?

A. The `guest` user is no longer supported by default in WebLogic Server 7.0. In WebLogic Server 6.x, `guest` was both the name of a user and the name for anonymous logins. The new user name for an anonymous user is `<anonymous>`. You can change this username by booting WebLogic Server with the following command line argument:

```
-Dweblogic.security.anonymousUserName=newAnonymousUserName
```

This argument allows you to make the name of the anonymous user `guest` for the purpose of backward compatibility.

If you want to have a `guest` user in WebLogic Server 7.0, create a user with the name `guest` in the Authentication provider in the default security realm and give the user the password of `guest`. If your code depends on the `guest` user, you should consider rewriting it to use the utility methods in `weblogic.security.WLSPrincipals`.

Q. I want to provide additional fields in my Web application for form-based authentication. What application programming interfaces (APIs) should I use?

A. The `CallbackHandler` implementation in the WebLogic Authentication provider supports only stringified versions of a username and password when using form-based authentication.

If a Web application requires more authentication information, use the `javax.security.auth.TextInputCallback` interface of the JAAS Callback application programming interface (API) in the code for your `LoginModule`. The implementation of the `javax.security.auth.TextInputCallback` interface needs to use the name of the authentication field as the prompt to the method constructor. For example:

```
Callback[] callbacks=new Callback[1];
callbacks[1]=new TextInputCallback("TextField");

try{
    callbackHandler.handle(callbacks)
    textField1=((TextInputCallback)callbacks[2]).getText
} catch (java.io.IOException ioe) {
    throw new LoginException(ioe.toString());
}catch (UnsupportedCallbackException uce) {
    throw new LoginException
        ("Error:"+uce.getCallback().toString() +
        "not available to garner authentication information" +
        "from the user");
}
//"textField1 is not set correctly
```

When the `ServletCallbackHandler` gets a `TextInputCallback`, the callback looks for a field matching the prompt of the `TextInputCallback`. If there is a match, the callback handler sets the value in the callback. If no match is found, an `UnsupportedCallback` exception is raised.

Q. I am using the 6.x security realm APIs in my application. How do I upgrade this functionality to the security APIs in WebLogic Server 7.0?

Specifically, I am using the `security.getRealm()` method and then the `getGroup()`, `getGroups()`, and `getUser()` methods on the returned realm.

A. You can use your 6.x application as is in the WebLogic Server 7.0 environment by using Compatibility security.

The management of WebLogic Server changed in 6.0 away from direct calls on the realm to the use of MBeans. This change was the first step in creating a unified administration model for WebLogic Server. While the Security MBeans introduced in 6.0 mirrored the capabilities defined for the realm, they were not flexible enough to allow security vendors to integrate their products with WebLogic Server. WebLogic Server 7.0 provides a set of Security Service Provider Interfaces (SSPI) and Security SPI MBeans that allow you to write custom security products for WebLogic Server. These MBeans can also be used to replicate functionality in an existing application so that the application can be used in the WebLogic Server 7.0 environment.

If you choose to upgrade your realm, you will need to write an implementation of the MBeans found in the `weblogic.management.security` package. These MBeans allow you to implement management methods that are specific to your 6.x realm. For a description of using the MBeans in the `weblogic.management.security` package, see [Developing Security Providers for WebLogic Server](#).

The following are some hints that may help you upgrade applications based on security realms to the security architecture available in WebLogic Server 7.0:

- If you are using the security realm to authenticate users, instead use the JAAS API to authenticate users.
- [Table 6-1](#) lists the mappings between the interfaces in the 6.x of the `weblogic.security.acl` package to the interfaces in the 7.0 of the `weblogic.management.security.authentication` package.

Note: The `weblogic.security.acl` package is deprecated in WebLogic Server 7.0.

Table 6-1 Interface Mapping

Methods in the 6.x weblogic.security.acl package	Corresponding Method in the 7.0 weblogic.management.security. authentication package
<code>newUser()</code>	<code>UserEditor.createUser()</code>
<code>deleteUser()</code>	<code>UserRemover.removeUser()</code>
<code>newGroup()</code>	<code>GroupEditor.createGroup()</code>
<code>deleteGroup()</code>	<code>GroupRemover.removeGroup()</code>
<code>Group.addMember()</code>	<code>GroupEditor.addMemberToGroup</code>
<code>Group.removeMember()</code>	<code>GroupEditor.removeMemberFromGroup()</code>
<code>Group.isMember()</code>	<code>GroupMemberLister.listGroupMembers()</code> or <code>SubjectUtils.isUserInGroup()</code>
<code>Group.members()</code>	<code>GroupMemberList.listGroupMembers()</code>
<code>userExists()</code>	<code>UserReader.isUser()</code>
<code>isGroupMember()</code>	<code>GroupReader.groupExists()</code> , <code>GroupReader.isMember()</code>

Q. Does WebLogic Server support Diffie-Hellman or DSS/DSA digital certificates?

A. No. The exportable version of WebLogic supports only 512 bit RSA with 40 bit RC4. Additionally, Web browsers do not support these types of certificates, and there are no commercial issuers for DSA certificates.

Q. Can a Weblogic Server deployment have one RSA and one non-RSA certificate?

A. No.

Q. Must we pay RSA licensing costs for non-RSA client code?

A. WebLogic Server has licensed RSA for SSL communications between WebLogic Server and WebLogic clients. When using WebLogic Server, no extra licensing for RSA is necessary, although different rules apply to VARs.

Q. How do I use Netscape security certificates with WebLogic Server?

A. Netscape stores the private and public keys together in a key-pair file which prevents you from separating the public and private keys. Therefore, you must generate another request for a certificate, not using Netscape utilities. You can use the Certificate Request servlet in WebLogic Server to generate a request for a new certificate.

Q. Where does WebLogic Server store private keys?

A. WebLogic Server 7.0 looks for the private key in the configured Keystore provider. Before a private key can be located in the Keystore provider, you need to perform the following configuration steps:

1. Configure the Keystore provider using the Administration Console.
2. Load the private key into the configured Keystore provider.
3. Specify the Server Private Key Alias and Server Passphrase attributes on the SSL protocol tab in the WebLogic Server Administration Console.

If the private key is not found, WebLogic Server looks in the file specified in the Server Key File Name attribute on the SSL tab under the Server node.

For more information, see [Managing WebLogic Security](#).

Note: The WebLogic Keystore provider is deprecated in WebLogic Server 7.0 SP01.

Q. How do I restrict access to servlets and JSPs?

The [Java Servlet API Specification v2.3](#), at <http://www.java.sun.com/products/servlet/download.html#specs>, allows you to declaratively restrict access to specific Servlets and JSPs using the Web Application Deployment descriptor. Section 13.3.2 of the specification has an example deployment descriptor that uses declarative security. For more information, see [Programming WebLogic HTTP Servlets](#). You can also specify roles for EJBs and Web applications through the Administration Console. For more information, see [Managing WebLogic Security](#).

Q. Can I use RSA encryption algorithms and javax.crypto.* API to build applications?

A. No. WebLogic's RSA license does not permit end-users to use RSA classes directly. You must obtain your own license for the encryption libraries from RSA.

Q. Can I use a JNDI Initial Context to pass the security credentials of a WebLogic Server user?

A. The ability to use JNDI to pass security credentials was deprecated in 6.1 of WebLogic Server. You can still use this method in WebLogic Server 7.0. However, BEA recommends using the Java Authentication and Authorization Service (JAAS) `runAs()` method rather than JNDI to associate a user with a security context. For more information, see [Programming WebLogic Security](#).

Q. Are WebLogic Server passwords secure?

A. The `config.xml` file no longer has clear text passwords. In place of clear text passwords, the `config.xml` file has encrypted passwords. You cannot copy encrypted passwords from one domain to another. Instead, you can edit the `config.xml` file and replace the existing encrypted passwords with clear text passwords and then copy the file to the new domain. The Administration Console will encrypt the passwords the next time it writes to the file.

Q. Why do I get a Java security permission error when starting and running WebLogic Server?

For example: `java.security.AccessControlException:access denied
(description of error)`

What should I do?

A. The `RecordingSecurityManager` utility can be used to detect permission problems that occur when starting and running WebLogic Server. The utility outputs permissions that can be added to your Java security policy file to resolve the permission problems that the utility finds. The `RecordingSecurityManager` utility is available from the following BEA [download page](#).

Q. Why do I get a certificate configuration error when I start the my Weblogic Server?

For example: `Alert> <WebLogicServer> <Security> configuration problem
with certificate file`

A. It is possible that you did not specify a `WL_HOME` relative file name in your SSL configuration files.

For more information, see [Managing WebLogic Security](#).

Q. Why can't I establish an outbound SSL connection when using the demonstration certificates?

A. When establishing an SSL connection, the subject DN of the digital certificate must match the host name of the server initiating the SSL connection. Otherwise, the SSL connection is dropped. If you use the demonstration certificates, the host names will not match. To avoid this situation, use the following command-line argument when starting WebLogic Server:

```
-Dweblogic.security.SSL.ignoreHostnameVerification=true
```

This argument disables the Hostname Verifier which compares the subject DN's and host names. This solution is recommended in development environments only. A more secure solution is to obtain a new digital certificate for the server making outbound SSL connections.

Q. Why do I get a configuration error when establishing an SSL connection to WebLogic Server?

For example: <WebLogic Server> <SSLListenThread listening on port 8802> Failed to connect to t3s://localhost:8802.

A problem with the configuration of the SSL protocol will also raise the following exception:

```
<java.io.IOException: Write Channel Closed, possible handshaking or trust failure>
```

A. By default, WebLogic Server contains a Hostname Verifier that compares the subject DN's of digital certificates and host names. When establishing an SSL connection, the subject DN of the digital certificate must match the host name of the server initiating the SSL connection. If you use the demonstration certificates the host names will not match. To avoid this situation, use the following command-line argument when starting WebLogic Server:

```
-Dweblogic.security.SSL.ignoreHostnameVerification=true
```

This argument disables the Hostname Verifier. This solution is recommended in development environments only. A more secure solution is to obtain a new digital certificate for your WebLogic client.

In WebLogic Server 7.0, WebLogic clients perform a trusted certificate authority check on the digital certificate for WebLogic Server. The client may reject the digital certificate of WebLogic Server if the certificate was not issued by a certificate authority trusted by the client. Previous versions of WebLogic Server did not perform this check.

Q. Why does my servlet return a no certificate message?

A. Unless WebLogic Server is configured to ask a client for its digital certificate during the SSL handshake (referred to as Two-Way SSL), WebLogic Server will not have the digital certificate. You get this error when a WebLogic servlet or JSP tries to perform peer validation on the client. Set the Client Certificate Enforced attribute on the SSL tab under the Server node to configure WebLogic Server to request client certificates.

Q. Why doesn't my Custom security provider show up (that is, it doesn't appear as a Configure a new *Security_Provider_Type* link) in the Administration Console?

A. Check to make sure the system administrator put the MBean JAR file (MJF) in the `lib/mbeantype` directory.

Q. Why do I get an exception when configuring a custom security provider through the WebLogic Server Administration Console?

The exception syntax is as follows:

```
java.lang.NoSuchMethodExcept: couldn't find getter for Name on
examples.security.providers.providertype.MBeanName
```

A. The `wlManagement.jar` and `wlSampleSecurityProviders.jar` need to be removed from the CLASSPATH in the startup script for WebLogic Server.

Q. Why can't the Windows NT security realm in the CompatibilityRealm authenticate my Windows NT user?

A. Your Windows NT user may need the Log on Locally advanced user right in the local Windows NT domain in order for the Windows NT security realm to properly authenticate the user.

7 FAQs: Upgrading

- What is Compatibility mode?
- How do I run WebLogic Server 7.0 in Compatibility mode when upgrading?
- What is two-phase deployment?
- Why do I keep getting a NoSuchMethodError when running an application on WebLogic Server 7.0?
- What is WebLogic Platform 7.0?
- If I am a WebLogic Server 7.0 GA customer, should I upgrade to WebLogic Platform 7.0 (upgrade from 7.0.0.0 to 7.0.0.1)?
- I am a WebLogic Server 7.0 GA customer. How do I upgrade to WebLogic Server 7.0.0.1?
- What WebLogic Server version will I see when I try to download 7.0.0.1?
- What about WebLogic Server 7.0 Service Pack 1?

Q. What is Compatibility mode?

A. When you upgrade to WebLogic Server 7.0 from WebLogic Server 6.x, Compatibility mode allows you to keep your 6.x configuration of users, groups, and ACLs.

Q. How do I run WebLogic Server 7.0 in Compatibility mode when upgrading?

A. WebLogic Server 7.0 recognizes your 6.x config.xml file and automatically runs in Compatibility mode.

Q. What is two-phase deployment?

A. WebLogic Server 7.0 uses two-phase deployment. In previous versions of WebLogic Server, when you deployed an application, a copy of the application file(s) was sent to all the targeted servers, which in turn, loaded the application. If any of those servers failed (or partially failed), the deployment was placed in an inconsistent state.

In the current release of WebLogic Server, the application is first prepared across the servers and then activated in a separate phase. In the prepare phase, the application is prepared for deployment without allowing user requests to the application. When the servers are ready, the application is activated everywhere. With this model, it is still possible to have a failure occur during the activation phase, which leads to the inconsistent state, but it is much less likely to occur.

For more information on this deployment model and other 7.0 deployment features, see [WebLogic Server Application Deployment](#).

Q. Why do I keep getting a `NoSuchMethodError` when running an application on WebLogic Server 7.0?

A. Make sure that you are not running the server with a patch for a previous version of WebLogic Server. For exact information of a particular patch, contact [Customer Support](#).

Also make sure that you do not have any JAR files in your server `CLASSPATH` that conflict with the version of J2EE that is supported by WebLogic Server 7.0, J2EE version 1.3. The 1.3 version of J2EE is added to the WebLogic Server 7.0 `CLASSPATH` by default. For example, you may have to remove `j2ee12.jar` from your server `CLASSPATH`.

Q. What is WebLogic Platform 7.0?

A. WebLogic Platform 7.0 is a BEA product that was released on June 28, 2002 and includes the following component products:

- WebLogic Server (WLS)
- WebLogic Portal (WLP)
- WebLogic Integration (WLI)
- WebLogic Workshop (WLW)

WebLogic Server and WebLogic Platform are at the 7.0.0.1 release level because WebLogic Server was released prior to the Platform, and then augmented to 7.0.0.1 to accommodate changes for the Platform release. WebLogic Platform reflects this WebLogic Server release level. Whenever a component product releases a Service Pack or a Rolling Patch, the WebLogic Platform release will be updated as well.

Q. If I am a WebLogic Server 7.0 GA customer, should I upgrade to WebLogic Platform 7.0 (upgrade from 7.0.0.0 to 7.0.0.1)?

A. Some reasons to upgrade or not to upgrade to 7.0.0.1:

You may want to upgrade to 7.0.0.1 because:

- you are using a layered product like WLI, WLP or WLW.
- you want to incorporate the fix for a CR that is fixed in 7.0.0.1.
- you are using the JAX-RPC API (Web Services) in WebLogic Server and want to use the implementation that is compatible with the finalized version of the spec (1.0).

You may not want to upgrade to 7.0.0.1 because:

- you could avoid upgrading twice by waiting for WebLogic Platform 7.0 Service Pack 1, which is tentatively set for the beginning of September 2002, and then upgrading from 7.0.0.0 to 7.0.1.0.
- you do not use any layered products.
- you want to wait for 7.0.1.0 so you can get incorporate even more fixes.

Q. I am a WebLogic Server 7.0 GA customer. How do I upgrade to WebLogic Server 7.0.0.1?

A. There is no **upgrade** installer, only **full** installers. Two ways to get to 7.0.0.1 are:

- From the Downloads section of www.bea.com, install WebLogic Server as part of the Platform installer. Choose **Custom** Installation, not **Typical** Installation, so that you can specify WebLogic Server only. Otherwise the full 250MB Platform will be installed.
- After logging on to [Customer Support](#), you can look under **BEA WebLogic Server Service Packs**. You will notice 7.0.0.1 under **WebLogic Server 7.0 Service Packs**.

Q. What WebLogic Server version will I see when I try to download 7.0.0.1?

A. C:\bea\weblogic700\samples\server\config\examples>java weblogic.version
WebLogic Server 7.0 Thu Jun 20 11:47:11 PDT 2002 190955
WebLogic XMLX Module 7.0 Thu Jun 20 11:58:44 PDT 2002 190955

You can differentiate from WebLogic Server 7.0 GA by date and change number (i.e. 190955).

Q. What about WebLogic Server 7.0 Service Pack 1?

A. WebLogic Platform 7.0 will be fully synchronized for WebLogic Server 7.0 Service Pack 1 (WLS, WLI, WLP, and WLW will all be delivered on the same date). The GA date for WebLogic Server 7.0 Service Pack 1 is still tentatively set for the beginning of September 2002. A smart **upgrade** installer that will recognize what you have installed and upgrade accordingly will be available with the WebLogic Platform 7.0 Service Pack 1 download.

8 FAQs: Web Services

- [Does WebLogic Server 7.0 support SOAP Messages with Attachments?](#)
- [Does WebLogic Server 7.0 support SOAP?](#)

Q. Does WebLogic Server 7.0 support SOAP Messages with Attachments?

A. Yes. As long as the data type of the attachment is on the list of supported JAX-RPC data types, then WebLogic Server will automatically handle parameters to Web services that are SOAP attachments. If you want to do further processing of the attachment, you can use handlers to intercept the request and response SOAP message (which includes the attachment.)

Q. Does WebLogic Server 7.0 support SOAP?

A. Yes. WebLogic Server's implementation of SOAP is included as part of the Web Services subsystem. For detailed information on creating Web Services with WebLogic Server, see [Programming WebLogic Web Services](#) at <http://e-docs.bea.com/wls/docs70/webServices/index.html>.

9 FAQs: Wireless-Related Questions

- What is a wireless (mobile) device?
- Can WebLogic Server support wireless devices?
- What should I consider when writing an application for wireless devices?
- What is WAP and what is i-Mode?
- Will my solution work on different networks such as CDMA, GPRS, TDMA and PDC-P?
- What changes do I need to consider for 3G wireless networks?

Q. What is a wireless (mobile) device?

A. A wireless device in this context is a device that has connectivity to the Internet without being physically plugged into a network with a wire. The most common examples of these are the Internet-enabled cell phone such as a WAP Phone or i-Mode phone, personal digital assistant (PDA) such as Palm VII, Pocket PC such as Wireless iPaq, and pager such as RIM Blackberry.

Q. Can WebLogic Server support wireless devices?

A. Yes. For information on wireless support, see *Using WAP with WebLogic Server*. Wireless examples, if installed, are located in the /samples/examples directory of your WebLogicServer installation and are available from the Start menu.

Q. What should I consider when writing an application for wireless devices?

A. The following is a list of factors to consider when writing applications for wireless clients:

- Content should be personalized based on the location of the device and/or the type of the device.
- Device microbrowsers are usually not HTML-based; some are WML-based, some are cHTML-based, some are HDML-based, some use Web Clipping, etc.
- Some devices have additional features such as Bluetooth, power keys and SMS messaging that can be used to enhance an application.
- Often the devices can be used with voice portals using voice recognition and text-to-speech.
- Most of the devices have smaller screens that may or may not display color or graphical images.
- These screens range in form from a vertically-oriented rectangle to a square to a horizontally-oriented rectangle.
- Many of the devices have a difficult data entry and selection mechanism with numeric keypads or styli.
- Connectivity is often lost when the device is mobile.
- The devices usually have no PKI security capability.
- A few of the devices are limited in the amount of data that can be transmitted to them for a page.

For more information see [Wireless, Internet and Email](#).

Q. What is WAP and what is i-Mode?

A. WAP and i-Mode are the two major over-the-air (OTA) protocols for wireless Internet communication with cell phones and some other devices. WAP stands for Wireless Application Protocol and is found predominantly in Europe and North America. i-Mode is the protocol used in Japan by NTT DoCoMo. A new protocol called WAP-NG (WAP Next Generation) is being considered as a replacement for both of these protocols.

Both WAP and i-Mode are comprised of both the OTA protocol and a markup language understood by their microbrowsers. The WAP markup language is WML (Wireless Markup Language) and cHTML (Compact HTML) is the markup language specified by i-Mode. WML and cHTML may be superseded by XHTML (Basic) in the future.

Note that other wireless carriers and devices use protocols and markup languages other than WAP and i-Mode. For example, the Palm VII uses web clipping over a proprietary protocol.

For more information about WAP, see <http://www.wapforum.org/faqs>.

Q. Will my solution work on different networks such as CDMA, GPRS, TDMA and PDC-P?

A. Yes. WAP and i-Mode were designed to hide the network details from the application developer. They will both work equally well on any underlying network. Therefore, as carriers upgrade their networks, applications written for either WAP or i-Mode will continue to work without any need for modifications. As the networks are upgraded to higher speeds, the performance of wireless applications written in either WML or cHTML should improve as well.

Q. What changes do I need to consider for 3G wireless networks?

A. None. As described in the previous answer, neither WAP nor i-Mode is dependent on the underlying network. However, a developer may consider enriching an application with streaming media or other content that requires greater bandwidth.

10 FAQs: XML

- Which XML parser comes with WebLogic Server 7.0?
- Is an XSLT processor bundled in WebLogic Server?
- What version of the JAXP API specification is implemented in WebLogic Server 7.0?
- Can I use the `getAttribute()` and `setAttribute()` methods of Version 2.3 of the Java Servlet API to parse XML documents?
- Can I plug in a version of Apache's Xerces XML parser that is different to the one that is built-in with WebLogic Server 7.0 (Xerces 1.4.4)?
- I plugged in a version of Apache Xalan that I downloaded from the Apache Web site, and now I get errors when I try to transform documents. What is the problem?
- How do I identify the document type of an XML document?

Q. Which XML parser comes with WebLogic Server 7.0?

A. We bundle the following two parsers with WebLogic Server 7.0:

- a built-in parser based on Apache's Xerces 1.4.4 parser
- WebLogic FastParser, a high-performance non-validating parser that you can use for small to medium sized XML documents.

The WebLogic XML Registry allows you to configure the parser you want to use for specific document types.

You can also use the WebLogic XML Streaming API to parse your XML documents. The Streaming API is based on the SAX API, but provides a more procedural, stream-based handling of XML documents rather than having to

write SAX event handlers, which can get complicated when dealing with complex XML documents.

Q. Is an XSLT processor bundled in WebLogic Server?

A. Yes, we bundle an XSLT processor, based on Apache's Xalan 2.2 processor, in WebLogic Server 7.0.

Q. What version of the JAXP API specification is implemented in WebLogic Server 7.0?

A. Version 1.1. This version includes pluggable XML transformation as well as pluggable XML parsing.

Q. Can I use the `getAttribute()` and `setAttribute()` methods of Version 2.3 of the Java Servlet API to parse XML documents?

A. Yes. Use the `setAttribute()` method for SAX mode parsing and the `getAttribute()` method for DOM mode parsing. Using these methods in a Servlet, however, is a WebLogic-specific feature. This means that the Servlet may not be fully portable to other Servlet engines, so use the feature with caution.

Q. Can I plug in a version of Apache's Xerces XML parser that is different to the one that is built-in with WebLogic Server 7.0 (Xerces 1.4.4)?

A. Yes. The version of Xerces that you can plug in, depends on the specific service pack of WebLogic Server 7.0 that you have installed.

If you have installed WebLogic Server 7.0.0.1 or lower on your computer, follow these instructions:

You can plug in the following versions of Xerces:

- Xerces 1.2.2
- Xerces 1.2.3
- Xerces 1.3.0
- Xerces 1.3.1
- Xerces 1.4.0
- Xerces 2.0.0
- Xerces 2.0.1

Warning: If you plug in version 2.0.0 or higher of Xerces, you cannot use the Xalan compatibility classes.

To plug in a different version of the Apache Xerces parser, follow these steps:

- a. Append the `xerces.jar` file to the end of WebLogic Server's CLASSPATH variable. This variable is set, by default, in the `WL_HOME/server/bin/startWLS.cmd` script used to start instances of WebLogic Server, where `WL_HOME` refers to the top-level directory of the WebLogic Platform.

You can also put the `xerces.jar` file in the `WEB-INF/lib` directory of the WAR file of your application. In this case, be sure you do not enable the `PreferWebInfClasses` flag for your Web application.

- b. Configure your XML Registry to use the `org.apache.xerces.jaxp` for the `DocumentBuilderFactory` or `SAXParserFactory` factories. For details, see [Administering WebLogic Server XML at `http://e-docs.bea.com/wls/docs70/xml/xml_admin.html`](http://e-docs.bea.com/wls/docs70/xml/xml_admin.html).

If you have installed WebLogic Server 7.0 Service Pack 1 on your computer, follow these instructions:

You can plug in *any* version of the Xerces XML parser.

To plug in a different version of the Xerces parser, follow these steps:

- a. Put the `xerces.jar` file in the `WEB-INF/lib` directory of your application. Be sure you enable the `PreferWebInfClasses` flag for your Web application.
- b. Configure the parser and builder factories through files in the `META-INF/services` directory in one of the archives in `WEB-INF/lib` if they are not already present in the `xerces.jar` file.

Warning: If you plug in a new version of Xerces using this procedure, you cannot use Administratin Console to configure the XML registry to configure the parser or entity resolution. If you need to configure the XML registry, use the procedure for plugging in new parsers for version 7.0.0.1 or lower described in the beginning of this section.

Q. I plugged in a version of Apache Xalan that I downloaded from the Apache Web site, and now I get errors when I try to transform documents. What is the problem?

A. You must ensure that the version of Apache Xalan you download from the Apache Web site is compatible with the version of Apache Xerces you are using with

WebLogic Server. This version is either the built-in version of 1.4.4 or one of the versions listed in the preceding question if you have plugged in your own version of Xerces.

Q. How do I identify the document type of an XML document?

A. If the XML document has a Public ID, then that is its document type. For example, if an XML document contains the following DOCTYPE declaration:

```
<!DOCTYPE mydoc PUBLIC "My public ID String"  
    "http://foo.com/url/to/my/dtd">
```

then its document type is My public ID String.

If the DOCTYPE declaration does not contain a Public ID, but specifies a System ID, then the document type is the System ID. For example, in the following DOCTYPE declaration:

```
<!DOCTYPE mydoc SYSTEM "http://foo.com/url/to/my/dtd">
```

the document type is `http://foo.com/url/to/my/dtd`.

Note: The System ID is of the DTD, not of the XML document itself. It can, however, still be used as a way to identify the XML document.

If the XML document does not specify a DOCTYPE declaration, then the document type can be either the root element name or the namespace URI, if it has one.