



# BEA WebLogic Server

## Frequently Asked Questions

BEA WebLogic Server 6.0  
Document Date: March 22, 2001

## Copyright

Copyright © 2001 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, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

All other product names are the property of companies with which they are associated.

### ***BEA WebLogic Server Frequently Asked Questions***

<b>Part Number</b>	<b>Document Date</b>	<b>Software Version</b>
N/A	March 22, 2001	BEA WebLogic Server Version 6.0

---

# Contents

## About This Document

Audience.....	xiv
e-docs Web Site.....	xiv
How to Print the Document.....	xiv
Related Information.....	xv
Contact Us!.....	xv
Documentation Conventions .....	xvi

## 1. FAQs: Administration and Configuration

What are safeguards for modifying the config.xml file? 1-1
How do I edit the config.xml file? 1-1

## 2. FAQs: Applets

What are the alternatives to using applets? 2-1
Can I use a “native” two-tier driver for a browser applet? 2-1
Why doesn’t my browser applet connect to the database? 2-2
Why does my applet work with Appletviewer, but not with a browser? 2-2
What’s causing ClassFormatErrors with my applet? 2-3

## 3. FAQs: Clustering

How do stubs work in a WebLogic Server cluster? 3-1
What happens when a failure occurs and the stub cannot connect to a WebLogic Server instance? 3-1
How does a server know when another server is unavailable? 3-1
How are notifications made when a server is added to a cluster? 3-2
How do clients learn about new WebLogic Server instances? 3-2

---

How do clients handle DNS requests to failed servers? 3-2

How many WebLogic Servers can I have on a multi-cpu machine? 3-2

Should I use a separate network for multicast in a cluster? 3-3

#### **4. FAQs: Code Examples**

Why can't I run an example class file with the java classname command? 4-1

How do I use the Cloudscape evaluation version with the EJB examples? 4-1

#### **5. FAQs: dbKona**

How are JDBC and dbKona related? 5-1

Does dbKona use vendor-specific native libraries to communicate with a remote DBMS? 5-1

How does performance compare between dbKona and raw SQL through JDBC? 5-2

What happens to a dbKona TableDataSet when I save changes? 5-2

#### **6. FAQs: EJB**

Why did my JDBC code throw a rollback SQLException? 6-1

Must my bean-managed persistence mechanism use the WebLogic JTS driver? 6-1

Why is there no polymorphic-type response from a create() or find() method? 6-1

Must EJBs be homogeneously deployed across a cluster? Why? 6-2

#### **7. FAQs: Installation**

How do I access and download Service Packs? 7-1

If I've already downloaded WebLogic Server, how can I upgrade when a new Service Pack is released? 7-2

How do I install Service Packs? 7-2

How do I view and recover files replaced or deleted by a Service Pack installation? 7-2

My Service Pack installation failed. Why did I receive this message: Java 1.3 and up is required? 7-2

How do I confirm that the Service Pack has been applied? 7-3

Can I apply the Service Pack to my Beta version of WebLogic Server 6.0? 7-3

---

Why isn't the Service Pack installer finding my installation of WebLogic Server?  
7-3

How do I download and install WebLogicServer 6.0 for Compaq Tru64 Unix  
using the WinZip file? 7-3

## **8. FAQs: Java**

Can you help me debug my program? 8-1

Where can I get help learning Java? 8-1

Where do I get the JDK? 8-1

How do I set up my CLASSPATH? 8-2

Why won't the examples work? 8-2

Where can I get help with Java error messages? 8-3

Why did a client-server message generate a StackOverflowException? 8-3

Will a JIT make my Java application run faster? 8-4

Can I redistribute the JDK that is bundled with WebLogic Server? 8-4

## **9. FAQs: WebLogic JDBC**

When should I use MultiPools? 9-1

How can I tell if a database is unavailable? 9-1

## **10. FAQs: JTA**

Can I use a non-XA driver in distributed transactions? 10-1

Can I use more than one non-XA connection pool in distributed transactions?  
10-2

How do XA and non-XA drivers differ in distributed transactions? 10-2

What XA drivers can I use in addition to the WebLogic jDriver for Oracle/XA?  
10-3

Can I use the Oracle thin driver as an XA driver in distributed transactions? 10-4

Do I need a 2PC licence when I use JMS with one JDBC non-XA driver? 10-4

Why am I getting an exception when I use JMS with a non-XA driver? 10-5

Why do I get an exception when I use EJB CMP 1.1? 10-5

Can I obtain a JDBC connection before I start a distributed transaction? 10-5

---

Can I close a JDBC connection after the distributed transaction is committed or rolled back? 10-6

## **11. FAQs: WebLogic jDriver for Informix**

How do I use multibyte character sets with WebLogic jDriver for Informix? 11-1

How do I allow multiuser access to my Informix database? 11-1

## **12. FAQs: WebLogic jDriver for Oracle**

Why does FOR UPDATE in Oracle 8 cause an **ORA-01002 error**? 12-2

What causes an OCIW32.dll error? 12-2

What transaction isolation levels does the WebLogic jDriver for Oracle support? 12-2

How do I use Unicode codesets with the WebLogic jDriver for Oracle driver? 12-3

How do I use OS Authentication with WebLogic jDriver for Oracle and Connection Pools? 12-3

What type of object is returned by ResultSet.getObject()? 12-4

How do I limit the number of Oracle database connections generated by WebLogic Server? 12-5

How do I call Oracle stored procedures that take no parameters? 12-5

How do I bind string values in a PreparedStatement? 12-5

Why do I get unexpected characters from 8-bit character sets in WebLogic jDriver for Oracle? 12-6

How do I learn what codesets are available in Oracle? 12-6

How do I look up an "ORA" SQLException? 12-7

What is error "ORA-6502?" 12-7

Why do I get an error while trying to retrieve the text for ORA-12705? 12-7

Why do I run out of resources during updates with Oracle's database link? 12-8

How do I prevent errors when running t3dbping? 12-8

## **13. FAQs: JMS**

What makes WebLogic JMS unique? 13-6

How has JMS changed between releases 5.x and 6.0? 13-7

---

What JMS features are new in WLS 6.1? 13-7

What problems were fixed in WLS 6.0 Service Pack 1 and what known problems still exist? 13-8

Can WLS 5.X JMS clients talk to WLS 6.X or vice versa? 13-8

Are there C/C++ interfaces to WLS JMS? 13-8

Is there a smaller version of weblogic.jar for supporting clients? 13-8

Where can I learn more about JMS? 13-8

What are the relationships between the various JMS objects in WLS 6.0? 13-9

How do I start WLS and configure JMS? 13-9

How do I configure JMS security? 13-10

Can I still use the default connection factories supported in WebLogic Release 5.1? 13-11

Why does `JMSSession.createTopic` or `JMSSession.createQueue` fail to create a destination in WLS JMS 6.0 (it worked in 5.1)? 13-12

How do I programmatically get a list of Queues or Topics? 13-13

How do I use a temporary destination? 13-13

How do I use MBeans to print runtime statistics? 13-14

Can two JMS servers share the same persistent store? 13-14

Which types of JDBC databases does WebLogic JMS support? 13-15

How do I use a third-party JDBC driver with JMS? 13-15

What if my JDBC database becomes corrupt? 13-16

How do I use persistence? 13-16

How does a file store compare with a JDBC store? 13-18

How do the WLS JMS 6.0 server/destination message maximum and threshold values work? 13-19

How do I configure JDBC so that the JMS JDBC Store recovers automatically? 13-19

Does WebLogic JMS support clustering? 13-20

How can I control on which WebLogic Server(s) my application will run? 13-21

What is the value of JMS clustering in WLS 6.0? 13-21

---

How do I perform a manual fail-over? 13-21

Does the WLS JMS server find out about closed or lost connections, crashes, and other problems and does it recover from them? 13-22

Do I need to use the WLS T3 protocol? 13-22

How do I do HTTP tunneling? 13-22

Does WLS JMS support SSL? 13-22

How do I integrate non-WLS JMS providers with WLS? 13-22

How do two-phase or global transactions relate to WebLogic JMS? 13-23

Why is my 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? 13-23

How can an application do a JMS operation and have it succeed, independent of the result of the transaction? 13-24

What happens if `acknowledge()` is called within a transaction? 13-25

Why do I get an error when using a non-transacted `TopicSession` from a transaction-required EJB? 13-25

Can I use a one-phase commit if my JMS JDBC store is on the same database for which I am doing other database work? 13-25

How do I integrate an `XAResource` with WLS to get JTA transactions with another resource manager? 13-26

Why do I get an exception when I start up JMS using an XA driver or with a TX data source? 13-26

Is WLS JMS `XAResource` compliant? 13-26

Why can't I receive a message that I send within a container-managed transaction? 13-27

What happens to a message that is rolled back or recovered? 13-27

Is it possible to set aside a message and acknowledge it later? 13-28

How should I use sorted queues? 13-29

How does sorting on message priority work? 13-29

How do I deal with a listener that doesn't keep up with messages being sent? 13-29

How do I get a thread dump to help track down a problem? 13-30



---

Do client identifiers need to be unique? 13-30

Are messages passed by copy/value or reference? 13-30

How do I manage a queue to view and delete specific messages? 13-31

How do I close a queue so that the messages will not be reloaded at the next server startup? 13-31

Why does an Object Message print as null after it is received? 13-31

In what order are messages delivered to a consumer? 13-31

Why do I get an exception when trying to find a connection factory? 13-32

Why should I avoid using selectors? 13-32

Is it possible to have multiple queue receivers listening on the same queue, using MessageSelectors (typically filtering on a correlation ID) to determine which listener actually receives the message? 13-33

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? 13-33

Why don't setting values work using `javax.jms.Message.setJMSPriority`, `DeliveryMode`, `Destination`, `TimeStamp` or `Expiration`? 13-33

What care must be taken when multi-threading JMS clients? 13-34

How should an application be set up to subscribe to multiple topics? 13-34

How should I use blocking and asynchronous `receive()` calls? 13-34

What precautions should I take when I use blocking `receive()` calls? 13-34

What is the `NO_ACKNOWLEDGE` acknowledge mode used for? 13-35

When should I use multicast subscribers? 13-35

When should I use server session pools and connection consumers? 13-36

How do I issue the `close()` method within an `onMessage()` method call and what are the semantics of the `close()` method? 13-36

How do I publish an XML message? 13-37

How do I use JMS in an applet? 13-37

How do I use a startup class to initialize and later reference JMS objects? 13-38

Is it possible to send or receive a message from within a message listener? 13-38

How do I create a producer pool? 13-41

---

What are pending messages in the console? 13-42

How do I use a less than or greater than on a message selector in ejb-jar.xml? 13-43

Is it better to have more or fewer sessions for a given number of subscribers? 13-43

Are foreign destinations handled within foreign JMS messages? 13-43

What is the standard way to create threads, do initialization, etc. within the application server? 13-43

Why do I get a JNDI problem when I name a Topic A.B and a second Topic A.B.C? 13-44

How many messages are sent across the network for processing topic messages? 13-44

What should an `XPATH` selector look like? 13-44

How do I handle request/response using JMS? 13-44

How do I put a message back on the queue for processing? 13-45

Is it OK to add new sessions and subscribers to a Queue or Topic Connection once it has been started? 13-45

What can I do when I get `java.lang.OutOfMemoryError` because producers are faster than consumers? 13-45

Why have different connection factories? 13-46

How should connections and sessions be allocated? 13-46

How does an application know if an application server goes down? 13-46

How do I debug WebLogic Server using Visual Cafe 4.1? 13-46

Is there a way to dynamically change an existing selector for a `TopicConsumer` using `setMessageSelect(String s)`? 13-47

How can I avoid asynchronous message deadlocks? 13-47

What are the advantages of message-driven beans? 13-48

How does concurrency work for message-driven beans? 13-49

Can an MDB be a message producer or both a producer and consumer? 13-49

If an MDB uses a durable subscription, will messages be accumulated if the MDB is not deployed? 13-50

---

How do I use non-WebLogic Server JMS provider destinations to drive MDB's? 13-50

Can you use a foreign JMS provider to drive an MDB transactionally? 13-50

How do I use JTA transactions within an MDB? 13-51

How do server session pools and message driven beans compare? 13-52

## **14. FAQs: Security**

How can I update my sample security certificates? 14-1

Why does the servlet return “no certificate”? 14-1

Does WebLogic support Diffie-Hellman or DSS/DSA digital certificates? 14-2

Can the server have one RSA and one non-RSA certificate? 14-2

Must we pay RSA licensing costs for non-RSA client code? 14-2

How do I use Netscape security certificates with WebLogic Server? 14-2

How do I restrict access to servlets and JSPs? 14-2

Can I use RSA encryption algorithms and `javax.crypto.*` API to build applications? 14-3

Can I use a JNDI Initial Context to pass the security credentials of a WebLogic Server user? 14-3

## **15. FAQs: Server-Related Questions**

How do I configure WebLogic to use a SOCKS proxy? 15-1

How can I speed up connection responses? 15-2

How does WebLogic support CORBA and client communication via IIOP? 15-2

How do I speed up HTTP tunneling? 15-2

Can WebLogic Server start with a UNIX boot? 15-3

What affects servlet performance besides clients and traffic? 15-3

Why do I get “NoClassDefFound”/“Too Many Open files” messages on Solaris? 15-3

How do I increase WebLogic Server memory? 15-4

What causes Java.io exceptions in the log file? 15-4

Java-CORBA integration: RMI-IIOP or Java IDL? 15-5

How do an RMI-IIOP application and an existing CORBA object interoperate?

---

15-5

What is the function of T3 in WebLogic Server? 15-5

How can I debug the Java code that I have running in WebLogic Server? 15-6

## **16. FAQs: Server-Side Java (Servlets)**

How do I call a servlet with parameters in the URL? 16-1

## **17. FAQs: XML**

Which XML parser comes with WebLogic Server 6.0? 17-1

Is an XSLT processor bundled in WebLogic Server? 17-1

Can I use SAX version 2.0 parsing with WebLogic Server 6.0? 17-1

Can I use the `getAttribute()` and `setAttribute()` methods of Version 2.2 of the Java Servlet API to parse XML documents? 17-2

Does WebLogic Server 6.0 support SOAP? 17-2

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 6.0 (Xerces 1.2.0)? 17-2

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? 17-2

---

# About This Document

This document introduces BEA WebLogic Server™ Frequently Asked Questions.

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 5, “FAQs: dbKona,” contains questions and answers about dbKona.

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 general questions and answers about Java.

Chapter 9, “FAQs: WebLogic JDBC,” contains questions and answers about WebLogic JDBC.

Chapter 10, “FAQs: JTA,” contains questions and answers about JTA.

Chapter 11, “FAQs: WebLogic JDBC for Informix,” contains questions and answers about WebLogic JDBC for Informix.

Chapter 12, “FAQs: WebLogic JDBC for Oracle,” contains questions and answers about WebLogic JDBC for Oracle.

Chapter 13, “FAQs: JMS,” contains questions and answers about JMS.

---

Chapter 14, “FAQs: Security,” contains questions and answers about security.

Chapter 15, “FAQs: Server-Related Questions,” contains general questions and answers about WebLogic Server.

Chapter 16, “FAQs: Server-Side Java (Servlets),” contains questions about servlets.

Chapter 17, “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](mailto: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>
UPPERCASE TEXT	Device names, environment variables, and logical operators. <i>Examples:</i> <pre>LPT1 BEA_HOME OR</pre>
{ }	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>



---

Convention	Usage
	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"><li>■ An argument can be repeated several times in the command line.</li><li>■ The statement omits additional optional arguments.</li><li>■ You can enter additional parameters, values, or other information</li></ul>
.	Indicates the omission of items from a code example or from a syntax line.

---



---

# 1 FAQs: Administration and Configuration

- What are safeguards for modifying the `config.xml` file?
- How do I edit the `config.xml` file?

**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.

**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/docs60/adminguide/cli.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/docs60/applets/usingapplets.html) at <http://e-docs.bea.com/wls/docs60/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 `jdbcDriver` 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 `jdbcDriver` 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?

**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.

First, it uses a heartbeat mechanism with an internal timeout to monitor the status of instances in a cluster. The heartbeat mechanism is based on the cluster's multicast address. Once a given server instance receives a set number of heartbeats, it is dropped from the cluster.

Second, WebLogic Server monitors socket errors to determine when server instances become unavailable. 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. For performance reasons, the number of WebLogic Servers should always be equal to or less than the number of processors (CPUs) on the machine.

For example, if the machine has four processors, the maximum number of WebLogic Servers on that machine is four. Running more servers than the number of processors on a single-machine could lead to software failure.

**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.



---

# 4 FAQs: Code Examples

- Why can't I run an example class file with the java classname command?
- How do I use the Cloudscape evaluation version with the EJB examples?

**Q.** Why can't I run an example class file with the java classname command?

**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. One of the first steps of almost every example is to make sure that your environment is set properly by running the setEnv script. This helps to ensure that your classpath is set properly and that you compile the class files into the appropriate directories. For more information, see the WebLogic Server Examples Guide located at *samples/examples/examples.html* in your WebLogic server distribution.

**Q.** How do I use the Cloudscape evaluation version with the EJB examples?

**A.** An evaluation version of the Cloudscape database management system is included in your WebLogic distribution, but must be set up properly for it to work with the code examples. All examples that use the Cloudscape database include instructions for configuring the connection pool. For additional information, see Using the Cloudscape Database With WebLogic Server located at *samples/eval/cloudscape/cloudscape.html* in your WebLogic Server distribution.



---

# 5 FAQs: dbKona

- How are JDBC and dbKona related?
- Does dbKona use vendor-specific native libraries to communicate with a remote DBMS?
- How does performance compare between dbKona and raw SQL through JDBC?
- What happens to a dbKona TableDataSet when I save changes?

**Q.** How are JDBC and dbKona related?

**A.** Java Database Connectivity (JDBC) is a JavaSoft specification that provides a standard way for programmers to write Java applications that access databases. Sun calls JDBC a [common base on which higher level tools and interfaces can be built](http://splash.javasoft.com/jdbc/index.html) at <http://splash.javasoft.com/jdbc/index.html>. *dbKona* is a higher-level interface than JDBC, but requires a JDBC driver, for example, one of the WebLogic jDriver JDBC drivers. dbKona provides convenient, vendor-independent ways to access data. It offers client-side management of query results and automatic SQL generation. dbKona objects shield the programmer from the necessity of knowing vendor-specific details in order to work with database data, yet offer the conformance and interoperability of the JDBC interface. With dbKona, for example, you can write a single Java application that can talk to multiple, heterogeneous databases.

Because dbKona is designed to be a plug-and-play API, which is how we support various DBMSes simultaneously, drivers from any vendor that meet the JDBC specification work with dbKona. Any application that uses dbKona will work with any JDBC driver, including WebLogic's JDBC drivers.

WebLogic jDriver products and dbKona give you the best of both worlds: complete compatibility with JDBC and the power of dbKona.

**Q.** Does dbKona use vendor-specific native libraries to communicate with a remote DBMS?

**A.** dbKona uses any JDBC driver to communicate with a remote DBMS. In the two-tier environment, if you use a Type 2 JDBC driver, you need a vendor-supplied library (or the proper ODBC driver) on each client using your dbKona application. Or you can use a Type 4 JDBC driver (pure Java) that does not require vendor libraries. In this three-layer arrangement, dbKona sits on top of the JDBC driver as a higher level interface. If you use a Type 2 JDBC driver, it makes calls to the vendor library to access the database.

The WebLogic pool, JTS, and RMI JDBC drivers make calls through the WebLogic Server to the JDBC driver on behalf of dbKona applications or applets, which do not need client-side vendor libraries.

**Q.** How does performance compare between dbKona and raw SQL through JDBC?

**A.** dbKona is very lightweight. It will cause little, if any, performance degradation, compared with the equivalent data retrieval in straight JDBC. Its advantages far outweigh its overhead: it provides a simpler and more convenient API for data access, and it gives you access to automatic generation of SQL through a query-by-example model. You can also do things with dbKona DataSets that you can't do with a straight JDBC ResultSet, like storing a dbKona TableDataSet in a WebLogic Server-side client Workspace.

Within WebLogic, your clients may actually *perceive* better performance, because dbKona gives you tools to manage query results. You can cache data on WebLogic for read-ahead. You can share one set of query results among multiple clients. With an "EventfulTableDataSet," you can even reflect changes to one client in all the clients viewing the same table. dbKona gives you more power over data than straight JDBC, so you can tune your client application for optimum performance.

**Q.** What happens to a dbKona TableDataSet when I save changes?

**A.** After you save changes to a TableDataSet, the results are left intact and you can continue to step through its records with the `fetchRecords()` methods.

You should be able to save the TableDataSet on a JDBC connection that is not the same JDBC connection you used to populate the TableDataSet with no effect on the integrity of the data in the TableDataSet. Of course, if the connection you use to save has some sort of history or state that makes it unusable for updates — for example, if you are in a transactional state that might cause the update to be uncommitted, or cause a deadlock because of locks other connections have — this will not be true. The DBMS must consider every connection as an independent application, even if the connections are cooperating on the client end.



---

# 6 FAQs: EJB

- Why did my JDBC code throw a rollback SQLException?
- Must my bean-managed persistence mechanism use the WebLogic JTS driver?
- Why is there no polymorphic-type response from a create() or find() method?

**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.

**Q.** Must my bean-managed persistence mechanism use the WebLogic JTS driver?

**A.** We recommend that you use the TxDataSource for bean-managed persistence.

**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.

---

# 7 FAQs: Installation

- How do I access and download Service Packs?
- If I've already downloaded WebLogic Server, how can I upgrade when a new Service Pack is released?
- How do I install Service Packs?
- My Service Pack installation failed. Why did I receive this message: Java 1.3 and up is required?
- How do I confirm that the Service Pack has been applied?
- Can I apply the Service Pack to my Beta version of WebLogic Server 6.0?
- Why isn't the Service Pack installer finding my installation of WebLogic Server?
- How do I download and install WebLogicServer 6.0 for Compaq Tru64 Unix using the WinZip file?

**Q.** How do I access and download Service Packs?

**A.** The latest WebLogic Server installers (with appropriate service packs already applied) are available at [http://commerce.bea.com/downloads/weblogic\\_server.jsp#wls](http://commerce.bea.com/downloads/weblogic_server.jsp#wls). The same URL includes a link to download Service Pack upgrades for an existing WebLogic Server installation (but these are available only to customers who have a WebSUPPORT account).

A list of bug fixes that are contained in the Service Pack, as well as other important information, is in the WebLogic Server 6.0 with Service Pack Release Notes. If you decide to apply the Service Pack, and something goes wrong, back-out of the installation by following the uninstall instructions.

**Q.** If I've already downloaded WebLogic Server, how can I upgrade when a new Service Pack is released?

**A.** Download only the new Service Pack by itself. Do not download a full instance of WebLogic Server with the service pack. To download a Service Pack by itself, if you have a WebSUPPORT account, you can go to <http://websupport.bea.com/custsupp>. If you do not have a WebSUPPORT account, go to our Web site at [http://commerce.bea.com/downloads/weblogic\\_server.jsp#wls](http://commerce.bea.com/downloads/weblogic_server.jsp#wls) and click the BEA WebSUPPORT link to get a WebSUPPORT account.

**Q.** How do I install Service Packs?

**A.** Click on one of the FTP links available on our Web site at [http://commerce.bea.com/downloads/weblogic\\_server.jsp#wls](http://commerce.bea.com/downloads/weblogic_server.jsp#wls) and save the Service Pack installer file to your hard drive. If you have a WebSUPPORT account, you can access links for service packs by themselves from <http://websupport.bea.com/custsupp>.

**Q.** How do I view and recover files replaced or deleted by a Service Pack installation?

**A.** To view the files replaced or deleted by a Service Pack installation, go to the `wls_6.0_prod_dir/uninstaller_servicepack` directory of the upgraded WebLogic Server instance and enter the following command:

```
jar tf baseRest.jar
```

To recover a file replaced or deleted by a Service Pack installation, go to the `wls_6.0_prod_dir/uninstaller_servicepack` directory of the upgraded WebLogic Server instance and enter the following command:

```
jar xf baseRest.jar filename
```

Where *filename* is the name of the file you want to recover.

**Q.** My Service Pack installation failed. Why did I receive this message: Java 1.3 and up is required?

WebLogic Server 6.0 Service Pack installer currently requires JDK 1.3 or higher. Make sure that you include the path to the JDK 1.3 software at the front of the PATH variable setting on your system. A 1.3.0 JDK is bundled with the server. For a Service Pack installation, point to that JDK. For example (using Windows):

```
set PATH=full_path_to_bea_home_dir\jdk130\bin;%PATH%
```

For example (using UNIX):

---

```
$PATH=full_path_to_bea_home_dir/jdk130/bin:$PATH export PATH
```

Where *full\_path\_to\_bea\_home\_dir* is the full pathname of the target BEA Home directory.

**Q.** How do I confirm that the Service Pack has been applied?

**A.** Check your server version with the command:

```
java weblogic.Admin -url host:port -username username  
-password password VERSION
```

Where *username* and *password* are your username and password.

This command will output version information. For example:

```
WebLogic Build: 6.0.1 Service Pack 1 ...
```

**Q.** Can I apply the Service Pack to my Beta version of WebLogic Server 6.0?

**A.** No. The Service Pack was designed only for the official release of WebLogic Server 6.0.

**Q.** Why isn't the Service Pack installer finding my installation of WebLogic Server?

**A.** First, make sure that you have the correct version of the WebLogic Server. If you run the VERSION command, it should return: WebLogic Build: 6.0. If you are certain you have WebLogic Server 6.0 installed, open the `registry.xml` file included in your BEA Home directory. The following code should be set:

```
<release level="6.0" Status="installed"
```

If you had installed the Beta version of WebLogic Server 6.0, you may see that the release level is "6.0 beta". You may also see that the Status is "uninstalled".

**Q.** How do I download and install WebLogicServer 6.0 for Compaq Tru64 Unix using the WinZip file?

**A.** Because WinZip files are typically extracted and executed, you should insert this file in a classpath like you would a .jar file.

Execute, do not unzip, this java-based installer.

```
java -classpath weblogic600_tru64.zip install
```



---

# 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 whitepapers 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 JDK130. 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/docs60/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.

Source	Description
<a href="http://developer.java.sun.com/developer/">Sun's Java Developer Connection at  http://developer.java.sun.com/developer/</a>	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.
<a href="http://www.mindprod.com/errormessages.html">Compiler Error Messages at  http://www.mindprod.com/errormessages.html</a>	An extensive list of compiler error messages, including the infamous NoClassDefFoundError.
<a href="http://java.sun.com/products/">Sun's Java APIs at  http://java.sun.com/products/</a>	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 one 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: WebLogic JDBC

- When should I use MultiPools?
- How can I tell if a database is unavailable?

**Q.** When should I use MultiPools?

**A.** If you are using WebLogic Server in a single-server configuration, you can use MultiPools in one of two ways 1) as a backup MultiPool in the event a database connection fails, or 2) as a 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 WebLogic JDBC Features](#) section of *Programming WebLogic JDBC* at <http://e-docs.bea.com/wls/docs60/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. For more information, read about Connection Pools and Multipools in the [Configuring WebLogic JDBC Features](#) section of *Programming WebLogic JDBC* at <http://e-docs.bea.com/wls/docs60/jdbc/programming.html>.



---

# 10 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?
- Why do I get an exception when I use EJB CMP 1.1?
- 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?

**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. (This configuration is the same as the JTS driver usage in WLS 5.1.)

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 calls `prepare` on it. When the Transaction Manager calls `commit` or `rollback` on it during the second phase, the WLS `XAResource` wrapper delegates the `commit/rollback` call to the non-XA JDBC connection. Any failure during `commit/rollback` results in heuristic exceptions. Application data may be left in an inconsistent state as a result of heuristic failure.

- **Redirecting Connections.** As in WLS 5.1, 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/docs60/adminguide/managetx.html>.

**Q.** Can I use the Oracle thin driver as an XA driver in distributed transactions?

**A.** Oracle 8.1.6 thin driver has a bug that does not accept any foreign Xid, and so does not work at all with any other vendor's transaction manager, including WLS.

Oracle 8.1.7 thin driver has threading problems and we do not recommend using it at this point. A workaround is not yet available in SP1. It will be available in the Silversword release. For this 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 (see FAQ 3 for description) 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.

Meanwhile, if you still want to try it out in SP1 without the workaround, you can configure the XA connection pool. For more information, see [Managing JDBC Connectivity](#) in the *Administration Guide*.

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. For more information, see [Limitations of the Weblogic jDriver for Oracle XA](#) in *Using WebLogic jDriver for Oracle/XA in Distributed Transactions*.

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.** Why do I get an exception when I use EJB CMP 1.1?

Problem: I am using distributed transactions with EJB CMP 1.1 and a non-XA connection pool. I configured the JDBC connection pool and the `TxDataSource` according to instructions in Can I use a non-XA driver in distributed transactions?. However, commit still gives `javax.transaction.xa.XAException: JDBC driver does not support XA, hence cannot be a participant in two-phase commit. Why?`

**A.** The old style CMP 1.1 DTD does not allow you to specify the data source name. When only the connection pool name is specified, the `EnableTwoPhaseCommit` setting of the `TxDataSource` is ignored. You should use the new style CMP 1.1 DTD and specify the data source name instead of the pool name.

To ensure that the descriptor is using the latest DTD file, verify that the DOCTYPE header for the WebLogic CMP 1.1 descriptor file is as follows:

```
<!DOCTYPE weblogic-rdbms-jar PUBLIC
'-//BEA Systems, Inc.//DTD WebLogic 6.0.0 EJB 1.1 RDBMS
Persistence//EN'
http://www.bea.com/servers/wls600/dtd/weblogic-rdbms11-persistence-600.dtd">
```

To see the DTD file, go to

<http://www.bea.com/servers/wls600/dtd/weblogic-rdbms11-persistence-600.dtd>.

**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 the distributed transaction is begun.

**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.

---

# 11 FAQs: WebLogic jDriver for Informix

- How do I use multibyte character sets with WebLogic jDriver for Informix?
- How do I allow multiuser access to my Informix database?

**Q.** How do I use multibyte character sets with WebLogic jDriver for Informix?

**A.** Currently, multibyte character sets are not supported for the WebLogic jDriver for Informix driver. Support may be added in a future release. For more information, read about codeset support in [Using WebLogic jDriver for Informix](http://e-docs.bea.com/wls/docs60/informix4/API_jinf4.html) at [http://e-docs.bea.com/wls/docs60/informix4/API\\_jinf4.html](http://e-docs.bea.com/wls/docs60/informix4/API_jinf4.html).

**Q.** How do I allow multiuser access to my Informix database?

**A.** If you get errors 271 (Could not insert new row into the table) and 243 (Could not position within a table table-name) when running a multiuser application using the WebLogic jDriver for Informix driver, first, confirm that your table file is not corrupted or truncated by using the `bcheck` or `secheck` utilities. If you have verified the file's integrity, then the most likely cause of your problem is that you have locked your table file or a record within the table. This is a normal event in multi user programs; generally you can recover by rolling back the current transaction, waiting a short time, and then re-executing the transaction. Here are some specific recovery procedures:

For interactive SQL, redo the operation.

- For C-ISAM programs, review the program logic and make sure that it can handle this situation. You can obtain exclusive access to a table by passing the `ISEXCLLOCK` flag to `ISOPEN`.
- For SQL programs, review the program logic and make sure that it can handle this situation. Use the statement `SET LOCK MODE TO WAIT`. For bulk updates,

see the `LOCK TABLE` statement and the `EXCLUSIVE` clause of the `DATABASE` statement.

---

# 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?

- How do I prevent errors when running t3dbping?

**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 website](http://www.unicode.org) at <http://www.unicode.org>. For a list of Unicode language abbreviations, see the [JavaSoft website](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 NT and UNIX, but not on Solaris. 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.

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 = OPS$
```

Note that the string “OPS\$” is arbitrary and up to the DBA.

2. Log in to the Oracle server as SYSTEM.
3. Create a user named OPS\$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 jDriver for Oracle always returns a Java object that preserves the precision of the data retrieved. WebLogic jDriver 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/docs60/jdbc.html) at <http://e-docs.bea.com/wls/docs60/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 tget 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, &acute;, &grave;, &acirc; 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='AL24UTF8SS' ;
```

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

```
VALUE
-----
AL24UTF8SS
```

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.** How do I prevent errors when running `t3dbping`?

**A.** When you are testing your Oracle database connections under UNIX, you can run `SQL*PLUS` and can successfully ping the database using `utils.dbping`. However, when you use the multitier `utils.t3dbping` utility, you receive an `ORA-12154` error message.

First, make sure that your `ORACLE_HOME` environment variable is correctly set to point to your Oracle installation. This variable *must* be set in the environment where the WebLogic server is running.

In the C-shell issue the following command:

```
$ setenv ORACLE_HOME path
```

where *path* is the path to your Oracle installation.

In the Bourne shell, issue the following commands:

---

```
$ ORACLE_HOME=path
$ export ORACLE_HOME
```

where *path* is the path to your Oracle installation. When you ping your database using the two-tier `utils.dbping` utility, the JDBC driver loads the database client library and establishes the connection to the database. When you use the multitier `utils.t3dbping` utility, the WebLogic Server loads a two-tier driver and uses it to establish a database connection. In both cases, the same method is used to connect to the database. SQL\*PLUS works because it doesn't require ORACLE\_HOME to find the client libraries.

If you are still experiencing problems, try this:

1. Open a command shell.
2. Run the two-tier version of `utils.dbping` in this shell.
3. Start WebLogic in this shell from the command line:  

```
$ java -ms32m -mx32m weblogic.server
```
4. Open a second command shell.
5. Run the `utils.t3dbping` in the second shell against the server running in the first command shell.

If this procedure doesn't work, please send the output from these commands to WebLogic technical support.





---

# 13 FAQs: JMS

## The WebLogic Server 6.0 Product

- What makes WebLogic JMS unique?
- How has JMS changed between releases 5.x and 6.0?
- What JMS features are new in WLS 6.1?
- What problems were fixed in WLS 6.0 Service Pack 1 and what known problems still exist?
- Can WLS 5.X JMS clients talk to WLS 6.X or vice versa?
- Are there C/C++ interfaces to WLS JMS?
- Is there a smaller version of weblogic.jar for supporting clients?
- Where can I learn more about JMS?
- What are the relationships between the various JMS objects in WLS 6.0?

## Configuration

- How do I start WLS and configure JMS?
- How do I configure JMS security?
- Can I still use the default connection factories supported in WebLogic Release 5.1?
- Why does `JMSSession.createTopic` or `JMSSession.createQueue` fail to create a destination in WLS JMS 6.0 (it worked in 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 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 WLS JMS 6.0 server/destination message maximum and threshold values work?
- How do I configure JDBC so that the JMS JDBC Store recovers automatically?
- Does WebLogic JMS support clustering?
- How can I control on which WebLogic Server(s) my application will run?
- What is the value of JMS clustering in WLS 6.0?
- How do I perform a manual fail-over?
- Does the WLS 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 WLS JMS support SSL?
- How do I integrate non-WLS JMS providers with WLS?

### Transaction Support

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

- 
- Why is my 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 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 JMS using an XA driver or with a TX data source?
  - Is WLS 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 selectors?
- Is it possible to have multiple queue receivers listening on the same queue, using MessageSelectors (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 don't setting values work using `javax.jms.Message.setJMSPriority`, `DeliveryMode`, `Destination`, `TimeStamp` or `Expiration`?
- What care must be taken when multi-threading 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 JMS in an applet?
- How do I use a startup class to initialize and later reference 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 JMS?
  - How do I put a message back on the queue for processing?
  - Is it OK 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?
  - How do I debug WebLogic Server using Visual Cafe 4.1?
  - 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 Server JMS provider destinations to drive MDB's?
- 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 [JavaSoft JMS specification version 1.0.2](#).
- 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.
- 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.
- Dependable customer support.

---

**Q.** How has JMS changed between releases 5.x and 6.0?

**A.** The following JMS features, which are new to WebLogic Server Release 6.0, are faster and more scalable than in 5.x:

- Clustering
- Two-phase transactions
- Multicasting
- File-based persistent message storage
- Quotas
- Queue sorting options
- Console-based configuration
- Extensible Markup Language (XML) message type
- `NO_ACKNOWLEDGE` acknowledge mode
- Simplified schema for JDBC persistent storage

**Q.** What JMS features are new in WLS 6.1?

**A.** The following features are new in WLS JMS 6.1.

- **Time To Deliver** - This feature allows producers of messages to associate a time to deliver with each message. A message with a time to deliver is essentially invisible until it is delivered - until the time to deliver. This allows users to schedule work at a particular time in the future.
- **Redelivery Delay** - Sometimes an external condition will prevent an application from handling a message. The application then rolls back or recovers the message, so that it may receive the message again later. If JMS immediately redelivers the message to the application the condition may not be clear and the application still may not be able to handle the message. This feature delays the redelivery of messages that are rolled back or recovered allowing time for the condition to clear so the application will have a better chance of handling the message.
- **Redelivery Limit** - This feature enables the messaging system to deal with problem messages. If a message is delivered to and returned by a consumer some user-specified number of times, then the message is deemed as

undeliverable and is redirected. Redirected messages can be sent to a configured `ErrorDestination`, or simply be dropped.

- Faster, in some cases significantly faster, than 6.0.

**Q.** What problems were fixed in WLS 6.0 Service Pack 1 and what known problems still exist?

**A.** See <http://developer.bea.com/docs/jms60sp1.jsp> for a paper on this topic.

**Q.** Can WLS 5.X JMS clients talk to WLS 6.X or vice versa?

**A.** No, this is not currently supported. At the base level, RMI is not interoperable. Above that, JMS uses substantially different protocols.

**Q.** Are there C/C++ interfaces to WLS 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 `weblogic.jar` for supporting clients?

**A.** There are two approaches to providing a thin client:

- A slimmed down version of `weblogic.jar` that supports enough to run the JMS client code - still uses RMI, etc.
- A very small client footprint that could run on J2ME (no RMI, serialization, etc.).

Neither is currently supported by WLS. There is discussion about how to generate your own slimmed down version of `weblogic.jar` in

<news://newsgroups.bea.com/3ad4ad17@newsgroups.bea.com>.

**Q.** Where can I learn more about JMS?

The following links provide more information about JMS:



---

## [Programming WebLogic JMS](#)

### [JavaSoft JMS specification version 1.0.2](#)

### [Ask BEA](#)

The JMS newsgroup `weblogic.developer.interest.jms` available from the `newsgroups.bea.com` server.

**Q.** What are the relationships between the various JMS objects in WLS 6.0?

**A.** The following relationships must be true:

- Multiple JMS Servers may exist on one WebLogic 6.0 server.
- A JMS connection factory can be deployed on multiple WebLogic servers within a cluster.
- Multiple consumers may consume from the same queue.
- Multiple producers may send to the same queue.
- A destination can be configured on only one JMS Server (the configuration may allow it on multiple JMS Servers but binding it into JNDI on the second server will fail at runtime).
- A JMS Server can be deployed on only one server.
- A queue exists only on the JMS Server it was configured for.
- A single JMS store can support multiple queues/topics.
- All the queues on one JMS Server share the same persistent store.
- No two JMS Servers can share the same persistent store. That is, a single JMS store can not support multiple JMS servers. There is a one-to-one relationship between JMS servers and stores.
- Multiple JMS stores may exist in a database, but they must each have a different table name prefix configured in order to differentiate them.
- There can be more than one persistent file store in a directory (see doc).

**Q.** How do I start WLS and configure JMS?

**A.** On NT, start WLS 6.X by selecting Start -> Programs -> BEA Weblogic E-Business Platform -> Weblogic Server 6.X -> Start Default Server and enter the administrator password.

On NT, to configure JMS, start the console by selecting Start -> Programs -> BEA Weblogic E-Business Platform -> Weblogic Server 6.X -> Start Default Console.

1. In the console tree view on the left, select JMS.
2. If you want persistent messages, first create a Store - Select Stores. In the window on the right, Select Create a new JMSFile Store for a file store, give it a name, give it a directory, select create. If you want a JDBCStore, you first need to create a JDBC connection pool by selecting JDBC in the tree view, Connection Pools, create a new JDBC Connection Pool. Select Targets, select a Target server, select the arrow that points to the right and select Apply. Then go back to Stores, Create a new JMSJDBCStore.
3. If you want to use a template, first create a Template - Select Templates. You need a template to create temporary queues. Select Create a new JMS Template, give it a name, select create, then you can move to the Thresholds & Quotas tab or the Override tab. Select Apply when done with your changes.
4. Select Servers. Select Create a new JMSServer, give it a name, select a Store if you created one, select a template if you created one, Select Create. Now you can move to the other tabs, make changes, select Apply. In particular, you must select Targets, select a Target server, select the arrow that points to the right, and select Apply. This is the server on which JMS will boot.
5. Create Destinations - from the tree view in the left panel, select the + in front of JMS, select the + in front of Servers, select the + in front of your server, select Destinations, Select Create a new JMSQueue or Create a new JMSTopic, fill in the first page and Select Create, then you can select, fill in, and Apply other tabs.
6. Create Connection Factories - on left tree view, open JMS. Select Connection Factory. Select Create a new JMS Connection Factory on the right panel. Type in the name and JNDI name. Select Create (lower right hand corner). Select the Targets tab. Select the name of the server on which you want to deploy the connection factory. Select the arrow pointing to the right - the server moves to chosen. Then select Apply (lower right hand corner).

**Q.** How do I configure JMS security?

---

**A.** The correct way to set up security for JMS is to go to the console, select ACLs in the tree view, then create some access control lists.

1. Set the ACL name which should be `weblogic.jms.queue.QUEUENAME` or `weblogic.jms.queue.TOPICNAME`.
2. Select Create.
3. Enter the New Permission of send or receive.
4. Select Create.
5. Enter a comma separated list of users or groups.
6. Select Grant Permission.
7. Select "saved to the realm implementation" to save your changes.
8. Select Yes.

This will update the `fileRealm.properties` file with lines that look like the following:

```
acl.send.weblogic.jms.queue.TestQueue1=user1
acl.receive.weblogic.jms.queue.TestQueue1=user1
```

If you don't have an ACL for a queue or topic, security is wide open.

There are also ACL's for accessing the JNDI context; the JNDI context is a requirement for initially accessing JMS. See the JNDI documentation.

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

**A.** Yes. The following two default connection factories have been deprecated:

```
javax.jms.QueueConnectionFactory
javax.jms.TopicConnectionFactory.
```

However, these connection factories are still defined and usable in this release for backwards compatibility.

WebLogic JMS 6.0 defines one connection factory, by default:

```
weblogic.jms.ConnectionFactory
```

You have to Enable the JMS default connection factories. Go to the console->your server->tuning->click on the checkbox Enable Default JMS Connection Factories.

You can also specify user-defined connection factories using the Administration Console.

**Q.** Why does `JMSSession.createTopic` or `JMSSession.createQueue` fail to create a destination in WLS JMS 6.0 (it worked in 5.1)?

**A.** In WLS 5.1 `createTopic()` or `createQueue()` creates the destination permanently in the database if it doesn't already exist, but does not modify the `weblogic.properties` file.

According to the [JavaSoft JMS specification version 1.0.2](#) regarding `createQueue()` and `createTopic()`, they are not for creating destinations dynamically. They are used to retrieve the destination referenced by using a string name instead of using JNDI lookup. The destination has to be in your `config.xml` file first. This change is documented in WLS 6.0 since it behaves differently than the previous release. You can use the WLS JMS helper class (`weblogic.jms.extensions.JMSHelper`) or the console to create destinations at the run time (note that there was a bug in 6.0 that caused a problem when the server restarted; this is fixed in Service Pack 1). These mechanisms create the destination and also modify the configuration file.

For more information on the `JMSHelper` classes, see the sub section called [Creating Destinations Dynamically](#) in *Programming WebLogic JMS*.

The following program creates a Topic.

```
import java.io.*;
import java.util.Hashtable;
import javax.jms.*;
import javax.naming.*;
import weblogic.jms.extensions.JMSHelper;

class t {
    public final static String
        JNDI_FACTORY="weblogic.jndi.WLInitialContextFactory";
    public final static String JMS_SERVER_NAME="TestJMSServer";
    public final static String DEST_JNDI_PREFIX="javax.destination.";

    static public void main(String [] args) throws Exception {
        try {
            Hashtable env = new Hashtable();
            env.put(Context.INITIAL_CONTEXT_FACTORY, JNDI_FACTORY);
            env.put(Context.PROVIDER_URL, "t3://localhost:7001");
            Context ctx = new InitialContext(env);
```

---

```
String topicName = "JMShelperTestQueue01";
String topicJNDI = DEST_JNDI_PREFIX + topicName;
System.out.println("topic name=" + topicName + ", jndi=" +
topicJNDI);
JMShelper.createPermanentTopicAsync(ctx, JMS_SERVER_NAME,
topicName,
topicJNDI);
} catch (JMSException e) {
e.printStackTrace();
}
}
}
```

**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 JMS Server where you want to be able to create temporary destinations. You can specify multiple JMS Server entries to support TemporaryTemplate and the system will load balance among those JMS Servers 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 JMS `Server` is defined something like:

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

After the template name, you can put pretty much 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](http://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 supports and provides JDBC drivers for the following databases:

- Cloudscape
- Informix
- Microsoft SQL (MSSQL) Server (Versions 6.5 and 7)
- Oracle (Version 8.1.6)
- Sybase (Version 12)

**Q.** How do I use a third-party JDBC driver with 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 Regeneration 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 Regeneration 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 1.0.2).

2. Make sure you are not using "Message.setJMSDeliveryMode". This is ignored, 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` 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 (client version 8.1.6 or later



---

is needed to run with WLS 6.0 on JDK 1.3). 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="TestTemplate1"
Targets="myserver" Store="JDBCStore">
<JMSQueue Name="TestQueue1"
JNDIName="jms.queue.TestQueue1"
Template="TestTemplate1"
/>
</JMSServer>
<JMSTemplate Name="TestTemplate1"
/>
<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 WLS JMS 6.0 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 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.** Several customers have reported a problem where they are using a JDBC store, the DBMS goes down and back up, but JMS can no longer use the store until WLS is shutdown and restarted. You can get around this problem by configuring the following attributes on the JDBC Connection Pool associated with the JMSJDBCStore:

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

If they are not set, then if the JDBC resource goes down and comes back up, JMS cannot re-use the connection pool until WLS is shutdown and restarted. This has been tested against WLS 6.0 SP02 and WLS 6.1.

**Q.** Does WebLogic JMS support clustering?

**A.** WebLogic JMS supports cluster-wide, transparent access to destinations from any server in the cluster. A system administrator can 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. Each connection factory can be deployed on multiple WebLogic Servers.

The application uses the Java Naming and Directory Interface (JNDI) to look up a connection factory and create a connection to establish communication with a JMS server. Each JMS server handles requests for a set of destinations. Requests for destinations not handled by a JMS server are forwarded to the appropriate server.

You can configure multiple JMS servers on the various nodes in the cluster as long as you give them different names. You can assign destinations to the various JMS servers.

One problem to be aware of is the propagation delay in replicating entries in JNDI. If you have an MDB deployed on one node but reference a destination on another node, the deployment may fail with a `javax.naming.NamingException` exception. The problem occurs because the server is not synced up to the JNDI from the remote server (JMS server) yet, so the JNDI lookup of destination as part of MDB deployment will fail. One workaround is for each MDB to reference a local destination. Another approach is deploy the MDB's after the server boots (plus a delay for JNDI propagation). To get around losing messages before the MDB is deployed, use durable subscribers. This problem is fixed for MDB's in WLS 6.1, where the MDB will be deployed and reconnection will be retried until the destination is available. Note that this is still a problem for EJB's in general that try to reference a non-local JMS destination.

Connection factories are described in more detail in [WebLogic JMS Fundamentals](#) in *Programming WebLogic JMS*. For more information on WebLogic Clusters, refer to [Using WebLogic Server Clusters](#).

---

**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.** What is the value of JMS clustering in WLS 6.0?

**A.** Consumers and producers can access a destination from any server in the cluster, independent of where the destination actually resides. This provides for location transparency. Note that it is necessary to configure the JMSServer on one of the members of the cluster (not the Admin Server) for this location transparency (which uses replicated JNDI).

If the server on which a given producer or consumer resides goes down, others can still access the destination. However, if the destination goes down, then all the producers and consumers are useless. If you lose a JMS server all transactions for which that server is enlisted are rolled back. JMS continues to work elsewhere in the cluster, just not on the server that went down. When that server comes back up then that server will again provide JMS service. When the server comes back up the messages are then pushed to or pulled by consumers as appropriate. Reboot the server and it should just work.

For example, assume a Topic is deployed on server B, a subscriber is accessing the Topic through server A, and a publisher is using server B. If server A goes down, the publisher can publish messages but the subscriber cannot receive messages (unless connecting to Server B).

In a future release, WLS JMS will handle automatic migration and fail-over when a server, in which a destination resides, goes down.

**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 WLS 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 WLS 6.0 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. Javasoft'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 WLS JMS support SSL?

**A.** Yes, SSL is supported in the WLS 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-WLS JMS providers with WLS?

**A.** See <http://developer.bea.com/docs/jmsproviders.jsp> for a discussion of integration with 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 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 (JMS spec 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 6.0 default factory `weblogic.jms.ConnectionFactory`, the session does not participate in a user transaction because the `UserTransactionsEnabled` flag is set to "False". If you use the deprecated default factory `javax.jms.QueueConnectionFactory` or `javax.jms.TopicConnectionFactory` 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 the 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 JMS JDBC store is on the same database for which I am doing other database work?

**A.** No. In WL6.0, 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 <http://developer.bea.com/docs/jmsjta.jsp> for a discussion of integrating MQ Series.

**Q.** Why do I get an exception when I start up 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 WLS JMS XAResource compliant?

**A.** WLS JMS does implement the XAResource interface as defined in the JTA specification. WLS 6.0 JMS does not implement XAConnection, XAConnectionFactory, XAQueueConnection, XAQueueConnectionFactory, XAQueueSession, XASession, XATopicConnection, XATopicConnectionFactory, or XATopicSession. These are defined as optional in the JMS specification. These methods are not methods that are 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 WLS 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, WLS JMS is automatically registered by WLS JTA so you don't need to worry about that.

It also means that you can't use WLS JMS with another transaction manager driving the transaction (at least, not with any documented interfaces).

---

These methods are available in WLS 6.1.

**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 requeued, then it would just sit there until there is one.

Starting in WLS 6.1, you can configure the maximum number of times that a message may be requeued. You can also specify a delay time before which the message will not be available after being requeued. If a message reaches its maximum requeue 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 (note that there is a known problem in WLS 6.0 trying to use a template to set the sort key).

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 "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 `/bea/wlserver6.0/config/mydomain/`):

```
java weblogic.Admin -url t3://localhost:7001 THREAD_DUMP
```

- From the console window, enter `ctrl-break` on NT.
- On Unix, kill 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 the 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 WLS JMS, you can administratively create and delete queues. There is no way to dynamically delete queues at run-time.

You could acknowledge the messages or not use persistence. You could use temporary queues. They exist only for the life of the connection. You could 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 Server 6.1 automatically deserializes the message if the application has done it before calling `toString()`.

**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 WLS 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 WLS 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 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 MessageSelectors (typically filtering on a correlation ID) to determine which listener actually receives the message?

**A.** The JMS Specification does not define the behavior. WebLogic Server JMS 6.0 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 have set your listener first and you will be first in the consumer list. However, each time an asynchronous queue consumer receives a message, the message 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 the JMS 1.0.2 specification, causing all messages currently in the subscription to be deleted.

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

**A.** These methods are for vendor use only. 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 JMS clients?

**A.** The [JavaSoft JMS specification version 1.0.2](#) 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. Note that unlike earlier versions of WebLogic Server JMS, 6.0 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:

- 
- 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()`.
  - 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.

**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 MDB's is preferable to using server session pools - see the question in this section, "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 checkbox when configuring the connection factory. For more information, see [JMS Connection Factories](#) in the *Administration Console Online Help*. If this checkbox 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 JMS in an applet?

**A.** This topic is covered in [news://newsgroups.bea.com/3ad321d5@newsgroups.bea.com](http://news://newsgroups.bea.com/3ad321d5@newsgroups.bea.com).

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

**A.** This topic is covered in [news://newsgroups.bea.com/3ad0d7f3@newsgroups.bea.com](http://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);
    }
    synchronzied(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 (new to WebLogic Server 6.1).
- subject to a delivery time (new to WebLogic Server 6.1) .

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 JavaSoft 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 Server 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 Server 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 is 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 Server 6.0 JMS, 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 Server 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 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 OK 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 the JMS 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. WLS 6.X does not support paging of messages out of memory.

**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. The 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-docs.bea.com/wls/docs60/javadocs/weblogic/jms/extensions/WLSession.html>

**Q.** How do I debug WebLogic Server using Visual Cafe 4.1?

**A.** You can install VisualCafe Enterprise Edition 4.1 and attach it to the server, pretty much as it worked for 3.X.

The following are steps for debugging using VC 4.1. You may change the directory names as necessary.

1. Install it under `d:\VisualCafeEE`. No special options are needed.

- 
2. Install the license under c:\\Program Files\\Common Files\\WebGain Shared.
  3. Start ddservices by selecting Start -> Programs -> WebGain Studio Professional -> Visual Cafe Enterprise Edition 4.1 -> Distributed Debugging Services -> Start DD Services (Java2 - 1.3)
  4. Start WebLogic Server using debugvm.exe instead of java.exe.  
cd D:\\bea\\wlserver6.0\\config\\mydomain  
setEnv  
edit startWebLogic.cmd  
change "%JAVA\_HOME%\\bin\\java" -hotspot -ms64m -mx64m to  
\\visualcafeEE\\jdk13\\bin\\debugvm.exe
  5. Run startWebLogic. It prints out some debugging information.
  6. Run VisualCafe - Start -> Programs -> WebGain Studio Professional -> Visual Cafe Enterprise Edition 4.1 -> Visual Cafe Enterprise Edition 4.1
  7. From the File menu, select Attach to Process. If everything is working correctly, you should see your machine name.
  8. Click the + sign to expand the tree and select your running WebLogic Server.

**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 the JMS 1.0.2 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.

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 [Using 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.

For Topics in WebLogic Server JMS 6.0, there is one `JMSSession` per bean instance in the pool. Because of the way Topics work, the session, and thus every bean instance, receives a copy of each message published on that Topic. There was also a problem that caused parallel processing not to work correctly. This has been fixed for WebLogic Server 6.0 Service Pack 1. 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 Server JMS provider destinations to drive MDB's?

**A.** See <http://developer.bea.com/docs/jmsmdb.jsp> for a white paper on this topic.

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

**A.** No. The message is asynchronously received outside a transaction and there is no J2EE API to then associate the message with a transaction.

The only reason this works for WebLogic Server JMS is that we have defined a WebLogic Server extension interface that has a method to associate a message with a transaction. This interface, `MDBTransaction`, is defined in [news://newsgroups.bea.com/3b3a009b\\$1@newsgroups.bea.com](http://news://newsgroups.bea.com/3b3a009b$1@newsgroups.bea.com). It has one method, `associateTransaction()`, that takes a `javax.jms.Message` parameter. This message must be associated with the transaction. We are hoping that other JMS vendors interested in integrating with WebLogic Server will implement this interface.

Another approach called source managed transactions, would be for there to be an API to tell the JMS provider to start a transaction on your behalf before delivering the message to an asynchronous consumer. This API doesn't exist in J2EE either. Even if there were such a provision, few non-WLS JMS providers can begin and drive such a transaction by themselves.

The current solution is to move all messages from the foreign destination to a WebLogic Server JMS destination (within a transaction if the foreign JMS provider supports it) and have that WebLogic Server JMS destination drive the MDB

---

transactionally (using the WLS JMS special interface). Currently, the messages can be moved between providers using code similar to that described in <http://developer.bea.com/docs/jmsproviders.jsp>. This code could be contained in a startup class that starts a thread and does the following:

```
while (true) {
start a transaction
receive a message synchronously with timeout
if timed_out { rollback and continue }
do the work
commit the transaction }
```

Doing a synchronous receive will have a problem in that the transaction may time out before the message is received. You can do the receive with a specified timeout, allowing enough time left in the transaction to complete the work. If the receive fails, roll back the transaction and try again. It is also not as efficient as MDB's since it does a synchronous instead of asynchronous receive (it ties up threads). Also, if you get a lot of timeouts, you will be burning more CPU.

Eventually, WLS JMS will provide a bridge to handle this message passing out-of-the box.

With any of these approaches, the `XAResource` must be registered with the Transaction Manager (this is done automatically for WebLogic Server JMS).

**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 rollback 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.



---

# 14 FAQs: Security

- How can I update my sample security certificates?
- Why does the servlet return “no certificate”?
- Does WebLogic support Diffie-Hellman or DSS/DSA digital certificates?
- Can the server 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?

**Q.** How can I update my sample security certificates?

**A.** To purchase your own certificates, you can generate a certificate signature request (CSR) using the Certificate Request servlet described in [Managing Security](#) in the *Administration Guide* at <http://e-docs.bea.com/wls/docs60/adminguide/cnfgsec.html>.

**Q.** Why does the servlet return “no certificate”?

**A.** You may receive this message when you try to use the certificate capture facility to provide two-way authentication. To troubleshoot, please check the following:

1. Does the Web browser you are using have a digital certificate?

2. Have you configured WebLogic Server to require client authentication by checking the Client Certificate Enforced option on the SSL tab in the Server Configuration window of the Administration Console?
3. Have you installed digital certificates in the `\wlserver6.0\config\mydomain` directory?
4. Have you configured a port for SSL communication as described in Configuring the SSL Protocol?
5. Are you using HTTPS rather than HTTP?

**Q.** Does WebLogic 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, browsers do not support these types of certificates, and there are no commercial issuers for DSA certificates.

**Q.** Can the server have one RSA and one non-RSA certificate?

**A.** No.

**Q.** Must we pay RSA licensing costs for non-RSA client code?

**A.** WebLogic has licensed RSA for SSL between WebLogic Servers and clients. With WebLogic, 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. In `config.xml`, the `ServerKeyFileName` attribute refers only to the private key file. Therefore, you must generate another request for a certificate, not using Netscape utilities. You can use the Certificate Request servlet to generate a request for a new certificate. Using WebLogic SSL shows how to use the servlet.

**Q.** How do I restrict access to servlets and JSPs?

**A.** The [Java Servlet API Specification v2.2](http://www.java.sun.com/products/servlet/download.html#specs), 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](#).

**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.** Yes, but not in releases that come after 6.1. In releases that come after 6.1, you must use the Java Authentication and Authorization Service (JAAS) rather than JNDI to associate a User with a security context. For more information, see [Programming WebLogic Security](#).



---

# 15 FAQs: Server-Related Questions

- 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?
- What causes Java.io exceptions in the log file?
- 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.** 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? at http://www.sunsite.unc.edu/javafaq/javafaq.html#proxy](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 /ORB support and not much on CORBA services. WebLogic supports CORBA in multiple ways.

First, Java clients can tunnel through a CORBA environment to WebLogic Server. We call this “IIOP tunneling,” and it is intended for use with applets coming through an IIOP firewall, such as the IONA Wonderwall product. This is a Java-to-Java model riding over an IIOP communications framework.

WebLogic RMI over IIOP provides RMI services for many clients (including CORBA clients) over IIOP. For more information, see [Using WebLogic RMI over IIOP at \[http://e-docs.bea.com/wls/docs60/rmi\\\_iiop/index.html\]\(http://e-docs.bea.com/wls/docs60/rmi\_iiop/index.html\)](http://e-docs.bea.com/wls/docs60/rmi_iiop/index.html).

WebLogic Enterprise Connectivity enables you to create IIOP connection pools to a BEA WebLogic Enterprise System, allowing you to execute WebLogic Enterprise CORBA objects from WebLogic Server servlets and Enterprise JavaBeans.

**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 at <http://www.eborder.nec.com/index2.htm> 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. Here is an example from an HP-UX 11 system, running under JDK 1.1. You need to supply the URL for your WebLogic Server and your system password. This file, `wlstart`, is placed in the `/sbin/init.d` directory and there is a link to it in the `/sbin/rc2.d` directory:

```
export SHLIB_PATH=\
/home/user1/weblogic/lib/hpux11:/oracle/8.0.4/lib
export CLASSPATH=/home/user1/weblogic/classes:\
/home/user1/weblogic/lib/weblogicaux.jar
export ORACLE_HOME=/oracle/8.0.4
export ORACLE_SID=DEMO
export ORACLE_TERM=vt100
export QAT=/home/user1/weblogic
cd $QAT
PATH=/sbin:/usr/sbin:/usr/bin:/opt/java/bin
export PATH
case $1 in
'start')
    java -ms64m -mx64m -verbosegc weblogic.Server > \
        /home/user1/weblogic/server.out 2>&1
    ;;
'stop')
    java weblogic.Admin URL shutdown system password
    ;;
*)
    echo "usage: $0 {start|stop}"
    ;;
esac
```

You should work with your UNIX system administrator to set up scripts similar to this for your system.

**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.) Start WebLogic Server with the **-ms32m** option to increase the allocation, as in this example:

```
$ java ... -ms32m -mx32m ...
```

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.** What causes Java.io exceptions in the log file?

**A.** You may see messages like these in the log file:

*(Windows NT)*

```
java.io.IOException Connection Reset by Peer
java.io.EOFException Connection Reset by Peer
```

*(Solaris)*

```
java.io.Exception: Broken pipe
```

These messages occur when you are using servlets. A client initiates an HTTP request, and then performs a series of actions on the browser:

1. Click Stop or enter equivalent command or keystrokes
2. Click Refresh or enter equivalent command or keystrokes

- 
3. Send a new HTTP request.

The messages indicate that WebLogic Server has detected and recovered from an interrupted HTTP request.

**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.

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.

**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 -iiop` against the interface to produce the stub for your RMI-IIOP client.
3. Run `rmic -idl` against the interface to produce IDL compatible with the RMI interface.
4. 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  
-Xrunjdw: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:

```
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.



---

# 16 FAQs: Server-Side Java (Servlets)

**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
}
```



---

# 17 FAQs: XML

- Which XML parser comes with WebLogic Server 6.0?
- Is an XSLT processor bundled in WebLogic Server?
- Can I use SAX version 2.0 parsing with WebLogic Server 6.0?
- Can I use the `getAttribute()` and `setAttribute()` methods of Version 2.2 of the Java Servlet API to parse XML documents?
- Does WebLogic Server 6.0 support SOAP?
- 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 6.0 (Xerces 1.2.0)?
- 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?

**Q.** Which XML parser comes with WebLogic Server 6.0?

**A.** We bundle a parser, based on Apache's Xerces 1.2.0 parser, in WebLogic Server 6.0. In addition, we include a parser generator that enables you to build a custom parser based on a specific DTD for optimized performance. The WebLogic XML Registry allows you to configure the parser you want to use for specific DTDs.

**Q.** Is an XSLT processor bundled in WebLogic Server?

**A.** Yes, we bundle an XSLT processor, based on Apache's Xalan 1.2 processor, in WebLogic Server 6.0.

**Q.** Can I use SAX version 2.0 parsing with WebLogic Server 6.0?

**A.** Yes, but you must use the parser specific API, instead of the Java API for XML Parsing (JAXP).

**Q.** Can I use the `getAttribute()` and `setAttribute()` methods of Version 2.2 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.** Does WebLogic Server 6.0 support SOAP?

**A.** WebLogic Server 6.0 does not support SOAP; however, BEA provides an alpha-quality SOAP implementation at <http://developer.bea.com:80/techtracks/xml/index.jsp>.

**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 6.0 (Xerces 1.2.0)?

**A.** No. Each version of Apache's Xerces XML parser requires a different version of the DOM interfaces and there can be only one copy of the DOM interfaces in the WebLogic Server environment. Version 6.0 of WebLogic Server includes the version of DOM needed for Xerces 1.2.0 because the built-in parser of WebLogic Server 6.0 is based on Xerces 1.2.0. If you try to plug in a different version, say 1.X of Xerces, you will get an error when you parse XML documents because Xerces version 1.X is expecting the version 1.X of the DOM interfaces. If you try to put version 1.X of the DOM interfaces in front of `weblogic.jar` in the CLASSPATH environment variable before you start WebLogic Server, you will cause WebLogic Server not to work correctly because Xerces 1.2.0 is used internally and requires the Xerces 1.2.0 version of the DOM interfaces.

The real cause of this problem is that Xerces ships new versions of interfaces in each release. Unfortunately, this prevents multiple versions of Xerces from co-existing in the same CLASSPATH environment.

The problem only affects Xerces or other parsers that support the non-final features of DOM Level 2. It does not affect the Sun Project X or Crimson parsers because those parsers support DOM Level 1 whose portion of the DOM interfaces is the same in all versions.

**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 Apache Xerces version 1.2.0. Because you cannot plug

---

in a different version of Apache Xerces (see the preceding question), the only version of Apache Xerces that is compatible with WebLogic Server 6.0 is 1.2.0.

The built-in parser (based on version 1.2.0 of Apache Xerces) and transformer (based on version 1.2 of Apache Xalan) have been modified by BEA to be compatible with each other.

