

# **BEA Weblogic Job Interview Questions And Answers**



**Interview Questions Answers**

**<https://interviewquestionsanswers.org/>**

## About Interview Questions Answers

**Interview Questions Answers . ORG** is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on BEA Weblogic will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [BEA Weblogic Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in BEA Weblogic category. To ensure quality, each submission is checked by our team, before it becomes live. This [BEA Weblogic Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.  
[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter for latest Jobs and interview preparation guides.  
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

**Interview Questions Answers.ORG Team**  
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



## BEA Weblogic Interview Questions And Answers Guide.

### Question - 1:

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

#### Ans:

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.

[View All Answers](#)

### Question - 2:

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

#### Ans:

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).

[View All Answers](#)

### Question - 3:

How do I look up an ORA SQLException?

#### Ans:

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
```

[View All Answers](#)

### Question - 4:

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

#### Ans:

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.

[View All Answers](#)

### Question - 5:

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

#### Ans:

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.

[View All Answers](#)

### Question - 6:

How do I protect WebLogic Server from security attacks from bogus clients using the WL-Proxy-Client-Cert header?

#### Ans:



The WL-Proxy-Client-Cert header can be spoofed (used) by any client which has direct access to WebLogic Server. WebLogic Server takes the certificate information from that header, trusting that it came from a secure source (the plug-in) and use that information to authenticate the user. In previous releases of WebLogic Server, the default behavior was to always trust that header. Now you need to explicitly define trust of the WL-Proxy-Client-Cert header. A new parameter clientCertProxy allows WebLogic Server to on the implicit trust of the certificate header. If you need an additional level of security, use a connection filter to limit all connections into WebLogic Server (therefore allowing WebLogic Server to only accept connections from the machine on which the plug-in is running).

The clientCertProxy parameter has been added to the HTTPClusterServlet and Web applications.

For the HTTPClusterServlet, add the parameter to the web.xml file as follows:

```
<context-param>
<param-name>clientCertProxy</param-name>
<param-value>true</param-value>
</context-param>
```

For Web applications, add the parameter to the web.xml file as follows:

ServletRequestImpl context-param

```
<context-param>
<param-name>weblogic.http.clientCertProxy</param-name>
<param-value>true</param-value>
</context-param>
```

You can also use this parameter in a cluster as follows:

```
<Cluster ClusterAddress="127.0.0.1" Name="MyCluster"
ClientCertProxyHeader="true"/>
```

[View All Answers](#)

### Question - 7:

How can I run multiple instances of the same servlet class in the same WebLogic Server instance?

**Ans:**

If you want to run multiple instances, your servlet will have to implement the SingleThreadModel interface. An instance of a class that implements the SingleThreadModel interface is guaranteed not to be invoked by multiple threads simultaneously. Multiple instances of a SingleThreadModel interface are used to service simultaneous requests, each running in a single thread.

When designing your servlet, consider how you use shared resources outside of the servlet class such as file and database access. Because there are multiple instances of servlets that are identical, and may use exactly the same resources, there are still synchronization and sharing issues that must be resolved, even if you do implement the SingleThreadModel interface.

[View All Answers](#)

### Question - 8:

How do I identify the document type of an XML document?

**Ans:**

If the XML document has a Public ID, then that is its document type. For example, if an XML document contains the following DOCTYPE declaration:

```
<!DOCTYPE mydoc PUBLIC "My public ID String"
"http://foo.com/url/to/my/dtd">
```

then its document type is My public ID String.

If the DOCTYPE declaration does not contain a Public ID, but specifies a System ID, then the document type is the System ID. For example, in the following DOCTYPE declaration:

```
<!DOCTYPE mydoc SYSTEM "http://foo.com/url/to/my/dtd">
```

the document type is http://foo.com/url/to/my/dtd.

Note: The System ID is of the DTD, not of the XML document itself. It can, however, still be used as a way to identify the XML document.

If the XML document does not specify a DOCTYPE declaration, then the document type can be either the root element name or the namespace URI, if it has one.

[View All Answers](#)

### Question - 9:

Which of the following are the benefits of MDB (Message Driven Beans) over standard JMS consumers?

**Ans:**

- a. In case of a MDB, developer needs to create a MessageListener class that utilizes a server-wide session pool.
- b. WebLogic Server container provides standard EJB services to MDBs.
- c. MDBs benefit from the write-once, deploy-anywhere paradigm of EJBs.
- d. MDBs can be associated with multiple Messaging Queues or Topics unlike standard JMS.

Choices B and C are correct. A message-driven bean is a special kind of EJB that acts as a message consumer in the WebLogic JMS messaging system. As with standard JMS message consumers, message-driven beans receive messages from a JMS Queue or Topic, and perform business logic based on the message contents. EJB deployers create listeners to a Queue or Topic at deployment time, and WebLogic Server automatically creates and removes message-driven bean instances as needed to process incoming messages.

Because message-driven beans are implemented as EJBs, they benefit from several key services that are not available to standard JMS consumers. Most importantly, message-driven bean instances are wholly managed by the WebLogic Server EJB container. Using a single message-driven bean class, WebLogic Server creates multiple EJB instances as necessary to process large volumes of messages concurrently. This stands in contrast to a standard JMS messaging system, where the developer must create a MessageListener class that utilizes a server-wide session pool. Thus choice A is incorrect.

WebLogic Server provides standard EJB services to MDBs, such as security services and automatic transaction management. Thus choice B is correct.

Being implemented as EJBs, MDBs benefit from the write-once, deploy-anywhere quality of EJBs. Whereas a JMS MessageListener is tied to specific session pools, Queues, or Topics, message-driven beans can be developed independently of available server resources. Thus Choice C is also correct.

It's not that MDBs are always advantageous as compared to standard JMS consumers. One limitation of MDBs compared to standard JMS listeners is that a given MDB deployment can be associated with only one Queue or Topic. If your application requires a single JMS consumer to service messages from multiple Queues or Topics, you must use a standard JMS consumer, or deploy multiple message-driven bean classes. Thus Choice D is incorrect.

[View All Answers](#)

### Question - 10:

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

**Ans:**



You can use tools such as WebGain, JBuilder, NetBeans and JDB that rely on the Java Platform Debugger Architecture (JPDA) to debug your Java code running in WebLogic Server.

JPDA is integrated in the Java 2 Platform, Standard Edition (J2SE) SDK 1.3 on all platforms and SDK 1.2.2 for Linux. There is a download available from Sun to add JPDA support to the J2SE SDK 1.2.2 on Solaris and Microsoft Window platforms. If you are using J2SE SDK 1.2.2 on these platforms you must first get this download.

To allow a debugger to attach to the virtual machine that WebLogic runs you have to start WebLogic in debug mode. In order to start WebLogic in debug mode using a Sun virtual machine follow these steps (start with step one only if using a Solaris platform):

1. If using a Solaris platform, change the LD\_LIBRARY\_PATH environment variable to prepend \$JAVA\_HOME/lib/sparc:

```
export LD_LIBRARY_PATH=$JAVA_HOME/lib/sparc:$LD_LIBRARY_PATH
```

2. Add the following parameters to the java command line (before the "weblogic.Server" string) that launches WebLogic server:

```
-Xdebug  
-Xnoagent  
-Xrunjdpw:transport=dt_socket  
server=y  
address=<port_for_debugger_to_connect>  
suspend=n  
-Djava.compiler=NONE
```

Note that with the Hotspot Performance engine the -Xnoagent and -Djava.compiler=NONE options are no longer required, but are accepted and ignored for compatibility reasons.

If server=y and no address parameter is supplied, WebLogic Server chooses the transport address and prints it to the standard output stream. So, if a line such as:

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.

[View All Answers](#)

### Question - 11:

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

**Ans:**

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 -iio 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.

[View All Answers](#)

### Question - 12:

What causes Java.io exceptions in the log file?

**Ans:**

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.

[View All Answers](#)

### Question - 13:

Considering the code below, which of the lines of code (given in the choices) should be placed at line 4 to make this JSP prints My score is : 100? Please ignore the line numbers for the purpose of validity of the JSP code.?

**Ans:**

1:  
2:  
3:

My Progress Report

4:

5: <% score++; %>

6: <%= "My score is : " + score %>

7:

8:

a. <%! int score = 99; %>

` b. <% int score; %>`

c. <%@ int score = 99; %>

` d. < int score = 99; />

A is the correct choice. The above JSP will work on declaring and initializing the variable score. The syntax for declaring and initializing a variable in JSP is as follows:

```
<%! variable=variable_value ; %>
```

Thus A is the correct choice. The <%@ ... %> tag is used to declare directives like include directive. Thus choice C is incorrect. The <% ... %> code is used to insert scriptlets (lines of code in java) like the one at line 5. The code written inside the scriptlet becomes part of the service() method of the generated servlet. Thus 'score' becomes the local variable of the service method. And for this JSP to compile properly, the variable 'score' should have been initialized. If "<% int score; %>" is



replaced by "<% int score=99; %>", the choice B would also be correct. In the present scenario, the choice B will give compilation error saying "Variable score may not have been initialized". Choice D is incorrect as it's not a valid tag.

[View All Answers](#)

### Question - 14:

Which is the only method defined in the javax.ejb.Handle interface?

**Ans:**

- a. getEJBHome
- b. getEJBObject
- c. getPrimaryKey
- d. getHomeHandle

Choice B is correct. The Handle is a serializable reference to the EJBObject. The EJBObject.getHandle() method returns a Handle object. The Handle allows us to recreate an EJB object remote reference that points to the same type of session bean or the same unique bean that the handle came from.

The Handle interface specifies only one method, getEJBObject(). Calling this method returns the EJB Object from which the handle was created. After getting the object back, we can narrow or cast it to the appropriate remote interface type. The getEJBHome method is defined in the HomeHandle interface and the getPrimaryKey method in the EntityContext interface. The getHomeHandle method is defined in the EJBHome interface.

[View All Answers](#)

### Question - 15:

A stateful session bean implementing the SessionSynchronization interface is deployed on the WebLogic server. Which of the following callback methods may be invoked on the bean?

**Ans:**

- a. beforeBegin
- b. afterBegin
- c. beforeCompletion
- d. afterCompletion

Choices B, C and D are correct. A stateful session bean using container-managed transactions can implement the javax.ejb.SessionSynchronization interface to provide transaction synchronization notifications. This interface defines only 3 methods - afterBegin(), beforeCompletion() and afterCompletion().

The afterBegin() callback method is called when the bean becomes part of a transaction. It is called before the EJB Object delegates the business method invocation to the bean instance. If the transaction is committed, the bean will be notified through its beforeCompletion() method. If the transaction is rolled back, this method is not invoked. The afterCompletion() method is always invoked whether the transaction ended with a commit or a rollback. A is incorrect because beforeBegin is not a method defined in the SessionSynchronization interface.

[View All Answers](#)

### Question - 16:

Can WebLogic Server start with a UNIX boot?

**Ans:**

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> #emp;
;;
'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.

[View All Answers](#)

### Question - 17:

How do I configure WebLogic to use a SOCKS proxy?

**Ans:**

You can configure a java.net socket to use SOCKS by setting a Java system property.

[View All Answers](#)



### Question - 18:

Which of the following programs can be created using the ZAC Publish Wizard tool?

**Ans:**

Choices:

- a. Installer
- b. Deployer
- c. Bootstrap
- d. Packager

Choices A and C are correct. To publish to a WebLogic Server, we can use the ZAC publish wizard. For this, the server must be running, and you will need to know a user and password that has permission to publish. You can also use the ZAC Publish Wizard to create a set of native programs - an installer and a bootstrap - for various operating systems that become part of a published Java application.

The installer program is a native executable that installs your published Java program on the local machine. It doesn't require a Java environment itself, so it can run out-of-the-box in the native OS. The bootstrap is also a native program; the user runs the bootstrap to invoke the published application. The bootstrap takes care of monitoring for updates, downloading and updating the user's application, and other administrative ZAC functions.

[View All Answers](#)

### Question - 19:

For EJB applications with bean-managed transaction demarcations, which of the following is used by the client to get a reference to the UserTransaction object for the WebLogic Server domain?

**Ans:**

- a. JTA
- b. JNDI
- c. JMS
- d. JTS
- e. JMX

Choice B is correct. WebLogic Server supports the javax.transaction package and the javax.transaction.xa package, which implement the Java Transaction API (JTA) for Java applications.

javax.transaction.UserTransaction provides an interface to the transaction manager that allows the application developer to manage the scope of a transaction explicitly. The client application uses JNDI to obtain an object reference to the UserTransaction object for the WebLogic Server domain.

The code used by the client is given.

```
UserTransaction ut=(UserTransaction)jndicontext.lookup("javax.transaction.UserTransaction")
```

If a bean needs a reference to the UserTransaction object, it obtains it from the EJBContext as given.

```
UserTransaction ut=ejbContext.getUserTransaction()
```

[View All Answers](#)

### Question - 20:

How do I enable debugging for the messaging bridge?

**Ans:**

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

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

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

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

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

```
DebugMessagingBridgeStartup="true"
```

```
DebugMessagingBridgeRuntime="true"
```

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

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

[View All Answers](#)

### Question - 21:

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

**Ans:**

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

[View All Answers](#)

### Question - 22:

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

**Ans:**

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

\* The supported adapters are located in the WL\_HOME\lib directory. For the Exactly-once QOS, the transaction adapter, jms-xa-adp.rar, must be deployed in the release 6.1 domain where the bridge is running, via the select Deployments -> Applications node on the console.

\* This jms-xa-adp.rar adapter must also be identified in the Adapter JNDI Name attribute as eis.jms.WLSCConnectionFactoryJNDIXA on the JMS Bridge Destination -> Configuration tab for both the source and target bridge destinations.

\* For WebLogic JMS, verify that you are using the transactional XAConnectionFactory for the queue or topic destinations mapped to both the source and target bridge destinations. To verify this, the following attributes must be set on the JMS -> Connection Factory -> Configuration -> Transactions console tab or in the configuration file (config.xml):

```
UserTransactionsEnabled=true
```

```
XAConnectionFactory=true
```





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

[View All Answers](#)

### Question - 23:

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

#### Ans:

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

\* Verify whether the bridge's source destination is correctly configured, by making sure that the following fields on the JMS Bridge Destination -> Configuration -> General console page have been properly completed:

- o Connection URL-this must be the URL of the JNDI provider used to look up the connection factory and actual destination.
- o Destination JNDI Name-this must be the JNDI name of the actual destination mapped to the source bridge destination.
- o Connection Factory JNDI Name-this must be the connection factory used to create a connection for the actual destination mapped to the source bridge destination.
- o User Name/Password-make sure that this user ID has permission to access the actual source destination.

\* Verify that the actual source queue or topic destination mapped to the source bridge destination is running and healthy, as follows:

- o Is the WebLogic Server instance hosting the source destination running?
- o Is the JMS server hosting the source destination correctly deployed?

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

[View All Answers](#)

### Question - 24:

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

#### Ans:

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](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 the "Using Foreign JMS Providers With WebLogic Server" white paper. 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 MDBs 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).

[View All Answers](#)

### Question - 25:

Which of the following are valid relationships between JMS objects in the WebLogic Server?

#### Ans:

- a. A single JMS store can support multiple JMS servers.
- b. Multiple consumers may consume from the same queue, but multiple producers may not send to the same queue.
- c. Multiple JMS Servers may exist on one WebLogic 6.0 server.
- d. A JMS Server can be deployed on only one server.

Choices C and D correct. JMS is an enterprise messaging system, which enables applications to communicate with one another through the exchange of messages. WebLogic JMS provides a full implementation of the JMS API. JMS objects are the objects necessary to connect to the JMS, and send and receive messages.

The major components of the WebLogic JMS Server architecture include WebLogic JMS servers implementing the messaging facility, Client applications, JNDI and Backing stores (file or database) for storing persistent data. Two or more JMS servers cannot share the same persistent store. 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. Multiple consumers may consume from the same queue and multiple producers may send messages to the same queue. Multiple JMS servers may exist on the same WebLogic server, but a JMS server can be deployed only on server at a time.

[View All Answers](#)

### Question - 26:

How can I avoid asynchronous message deadlocks?

#### Ans:

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.

[View All Answers](#)

### Question - 27:

Which of the following is NOT true about deploying EJBs in the WebLogic Server?

**Ans:**

- a. The weblogic/config/examples/applications directory acts as an automatic deployment directory for EJB 2.0 .jar files and EJB .jar deployment directories
- b. The automatic redeployment feature of the WebLogic Server can only redeploy an EJB's implementation classes, you cannot redeploy an EJB's public interfaces
- c. Before deploying a packaged jar file containing uncompiled EJB classes and interfaces, we have to use weblogic.ejb on the packaged .jar file to generate WebLogic Server container classes.

Choice C is correct because it is NOT true. A and B are true about deploying EJBs in the WebLogic server. The weblogic/config/examples/applications directory acts as an automatic deployment directory for EJB 2.0 .jar files and EJB .jar deployment directories. When you start WebLogic Server, it automatically deploys any valid EJB 2.0 .jar files or .jar directories that reside in the applications directory. WebLogic Server also checks the contents of applications every ten seconds to determine whether an EJB deployment has changed. If a deployment has changed, it is automatically redeployed using the dynamic deployment feature.

If you change the contents of a compiled EJB .jar file in applications, WebLogic Server automatically attempts to redeploy the .jar using the dynamic deployment feature. Since the automatic redeployment feature uses dynamic deployment, WebLogic Server can only redeploy an EJB's implementation classes. You cannot redeploy an EJB's public interfaces. You create compiled EJB 2.0 .jar files by Compiling your EJB classes and interfaces, packaging the EJB classes and interfaces into a valid .jar file and then Using weblogic.ejb on the .jar file to generate WebLogic Server container classes. An uncompiled EJB .jar file has the same structure as a compiled file, but you do not have to compile individual class files and interfaces and you do not have to use weblogic.ejb on the packaged .jar file to generate WebLogic Server container classes. So C is not true.

[View All Answers](#)

### Question - 28:

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

**Ans:**

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.

[View All Answers](#)

### Question - 29:

How do I handle request/response using JMS?

**Ans:**

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.

[View All Answers](#)

### Question - 30:

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

**Ans:**

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

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

In WebLogic JMS 6.1, 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.



[View All Answers](#)

### Question - 31:

Are foreign destinations handled within foreign JMS messages?

**Ans:**

WebLogic Server JMS does not know what to do with foreign destinations that it runs into. This issue has been discussed with Sun and the JMS 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).

[View All Answers](#)

### Question - 32:

Match the EJB functions given below with the functionality equivalent in SQL?

**Ans:**

- A.) ejbStore() 1.) INSERT
- B.) ejbLoad() 2.) UPDATE
- C.) ejbCreate() 3.) SELECT
- a. A->1, B->2, C->3
- b. A->2, B->1, C->3
- c. A->3, B->2, C->1
- d. A->1, B->3, C->2
- e. A->2, B->3, C->1
- f. A->3, B->1, C->2

Choice E is correct. When the create() method on a home interface is invoked, the container delegates the create() method call to the bean instance's matching ejbCreate() method. The ejbCreate() methods are used to initialize the instance state before record is inserted into the database. The ejbCreate() method is analogous to INSERT. The ejbStore() method is invoked just before the container is about to write the bean container-managed fields to the database. It is analogous to the UPDATE. The ejbLoad() is invoked just after the container has refreshed the bean container-managed fields with its state from the database. It is analogous to the SELECT. Thus choice E is correct and others are not.

[View All Answers](#)

### Question - 33:

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

**Ans:**

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>
```

[View All Answers](#)

### Question - 34:

How do I create a producer pool?

**Ans:**

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

```
class ProducerPool {
    static Hashmap pSets = new Hashtable();
    static Hashmap inUse = new Hashtable();
    QueueSender get(String contextURL,
String connectionFactoryName,
String destinationName) {
    String lookup = contextURL+";"
"+connectionFactName+";"+destName;
    synchronized(pSets) {
        producer set = pSets.get(lookup);
        if (set != null && set not empty)
            qs = set.removeFirst();
        }
        if (producer == null) {
            create ctx
            get connect factory
            create connection
            create session
            look up destination
            qs = create queue sender
        }
        synchronized(inUse) {
            inUse.put(qs, lookup);
        }
        return qs;
    }
    void put(QueueSender qs) {
        String lookup;
        synchronized(inUse) {
```



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

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

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

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

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

[View All Answers](#)

### Question - 35:

A client wants to preserve the reference to the EJBHome object of an enterprise bean instance and use it later. Which of the following can be serialized for this purpose?

**Ans:**

- a. Home
- b. Handle
- c. HomeHandle
- d. EJBHomeHandle
- e. HomeObject

Choice C is correct. Once a client has obtained the EJBHome object for an EJB instance, it can create a reference to the home object by calling getHomeHandle(). getHomeHandle() returns a HomeHandle object, which can be used to obtain the home interface to the same EJB instance at a later time.

A client can pass the HomeHandle object as arguments to another client, and the receiving client can use the handle to obtain a reference to the same EJBHome object. Clients can also serialize the HomeHandle and store it in a file for later use. The HomeHandle interface has only one method getEJBHome(), which returns the EJBHome reference.

[View All Answers](#)

### Question - 36:

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

**Ans:**

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

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

- \* Terminates the receipt of all pending messages. Applications may return a message or null if a message was not available at the time of the close.
- \* Waits until all message listeners that are currently processing messages have completed (except for the message listener from which the close() method is being called).
- \* Rolls back in-process transactions on its transacted sessions (unless such transactions are part of an external JTA user transaction).
- \* Does not force an acknowledgment of client-acknowledged sessions. By not forcing an acknowledgment, 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.

[View All Answers](#)

### Question - 37:

When should I use multicast subscribers?

**Ans:**

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.

[View All Answers](#)



### Question - 38:

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

#### Ans:

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

- \* Pass a timeout value as an argument to the receive() method and set it to the minimum value greater than zero, that is allowed by the application to avoid consuming threads that are waiting for a response from the server.

- \* Use the receiveNoWait() method which returns the next message or a null value if no message is currently available. In this case, the call does not block. The servlet should provide a way to return to or reschedule the request, without calling wait().

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

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

[View All Answers](#)

### Question - 39:

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

#### Ans:

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.

[View All Answers](#)

### Question - 40:

How does sorting on message priority work?

#### Ans:

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

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

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

[View All Answers](#)

### Question - 41:

Which of the following attributes in the Monitoring tab for a JDBC connection pool in the Administrative console tell us how many clients are currently waiting for a connection?

#### Ans:

- a. Waiters high
- b. Waiters
- c. Connections high
- d. Clients
- e. Wait seconds high

Choice B is correct. JDBC subsystem resources can also be monitored via the Administration Console. The Monitoring tab for a JDBC connection pool allows you to access a table listing statistics for the instances of that pool. These attributes provide important information for managing client database access.

The Waiters High field indicates the highest number of clients waiting for a connection at one time. The Waiters field tells you how many clients are currently waiting for a connection. The Connections High field indicates the highest number of connections that have occurred at one time. The Wait Seconds High field tells you the longest duration a client has had to wait for a database connection. These attributes allow you to gauge the effectiveness of the current configuration in responding to client requests.

[View All Answers](#)

### Question - 42:

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

#### Ans:

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.

[View All Answers](#)

### Question - 43:

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

#### Ans:

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 6.1 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.

[View All Answers](#)

### Question - 44:

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?

#### Ans:

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 thread 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.

[View All Answers](#)

### Question - 45:

How do I do HTTP tunneling?

#### Ans:



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 `<http>` 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).

[View All Answers](#)

### Question - 46:

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

#### Ans:

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.

[View All Answers](#)

### Question - 47:

How does a file store compare with a JDBC store?

#### Ans:

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.

[View All Answers](#)

### Question - 48:

Which of the following algorithms is used by the WebLogic Server as the default load balancing strategy for clustered object stubs when no algorithm is specified ?

#### Ans:

- Round-robin
- Weight-based
- Random
- None of the above

Choice A is correct. The basic idea behind load balancing is that by distributing the load proportionally among all the servers in the cluster, the servers can each run at full capacity. WebLogic Server clusters support several algorithms for load balancing clustered objects. The particular algorithm you choose is maintained within the replica-aware stub obtained for the clustered object. Configurable algorithms for load balancing clustered objects are: Round-robin, Weight-based and Random.

WebLogic Server uses the round-robin algorithm as the default load balancing strategy for clustered object stubs when no algorithm is specified. Round-robin is the only load balancing strategy used by WebLogic proxy plug-ins for HTTP session state clustering. The round-robin algorithm cycles through a list of WebLogic Server instances in order. For clustered objects, the server list consists of WebLogic Server instances that host the clustered object. For proxy plug-ins, the list consists of all WebLogic Servers that host the clustered servlet or JSP.

[View All Answers](#)

### Question - 49:

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

#### Ans:

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

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

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

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

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

Follow the procedures in JDBC Database Utility in Programming WebLogic JMS to manually create the database tables for the JDBC store.

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

[View All Answers](#)

### Question - 50:

Can two JMS servers share the same persistent store?

#### Ans:

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.





[View All Answers](#)

### Question - 51:

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

#### Ans:

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());
}
```

[View All Answers](#)

### Question - 52:

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

#### Ans:

Yes. The following two names for the 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 6.1 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 check box Enable Default JMS Connection Factories.

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

[View All Answers](#)

### Question - 53:

How do I start WLS and configure JMS?

#### Ans:

On Windows, 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 Windows, 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).

[View All Answers](#)

### Question - 54:

An instance of stateful session EJB when accessed simultaneously from more than one clients on same VM results in RemoteException or EJBException. In case the client is a Servlet thread, which of the following techniques can be used to avoid RemoteException/EJBException?

#### Ans:

a. Not possible.

b. Store the reference to the EJB instance as an instance variable of Servlet class.

c. Store the reference to the EJB instance as a local variable of Servlet class.

d. Make the Servlet client to be remote instead of internal to WebLogic server.

Choice C is the correct choice. An instance of a stateful session EJB can be accessed from only one client virtual machine at a time. Multiple client threads from the same virtual machine can access the same instance of a stateful session EJB, but they must do so in a serial fashion. If a client-invoked business method is in progress on an instance when another client-invoked call, from the same or different client, arrives at the same instance of a stateful session bean class, the container may throw the java.rmi.RemoteException to the second client, if the client is a remote client, or the javax.ejb.EJBException, if the client is a local client. Thus choice D is incorrect.

To avoid any exception, each Servlet should store a reference to a particular EJB instance in a local variable of the Servlet's service() method. Please note that





variables local to methods like service(), doGet(), doPost() are not shared between different requests and are automatically thread safe. Thus choice C is correct. An instance variable unlike local variable is shared. Thus Choice B is incorrect.

An implication of this rule is that an application cannot make loop back calls to a session bean instance.

This restriction does not apply to a stateless session bean because the container routes each request to a different instance of the session bean class.

[View All Answers](#)

### Question - 55:

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

**Ans:**

According to Oracle Metalink Bug Database Doc ID: 144784.1, in the absence of explicit data typecasting, OCI assumes that a bind variable is a CHAR data type. If the SQL statement intends to use the bind variable as a DATE data type, but OCI thought it was a CHAR, the SQL parser will have a conflict in data types. The fix is to explicitly use data conversion functions to convert the bind variables in the problem queries. For example, a select string of

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

should be changed to:

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

[View All Answers](#)

### Question - 56:

How many deployment descriptor files does a CMP entity bean deployed on the WebLogic Server have?

**Ans:**

- a. One J2EE specific deployment descriptor and two WebLogic specific deployment descriptors
- b. One J2EE specific deployment descriptor and one WebLogic specific deployment descriptors
- c. One J2EE specific deployment descriptor only
- d. One WebLogic specific deployment descriptor only

Choice A is correct. Deployment descriptors are text documents formatted with XML tags. The J2EE specifications define standard, portable deployment descriptors for J2EE components and applications. BEA defines additional WebLogic-specific deployment descriptors required to deploy a component or application in the WebLogic Server environment.

When packaging an enterprise bean, we need to create an ejb-jar.xml deployment descriptor in the META-INF subdirectory and add entries for the bean. We also need to create a weblogic-ejb-jar.xml deployment descriptor in the META-INF subdirectory and add entries for the bean. If the bean is an entity bean with container-managed persistence, first we create a weblogic-rdbms-cmp-jar-bean\_name.xml deployment descriptor in the META-INF directory with entries for the bean. Then we map the bean to this CMP deployment descriptor with a attribute in the weblogic-ejb-jar.xml file.

[View All Answers](#)

### Question - 57:

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

**Ans:**

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, Å, Ä, Åç would all map to a). Other characters are mapped to a question mark (?).

[View All Answers](#)

### Question - 58:

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

**Ans:**

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,

[View All Answers](#)

### Question - 59:

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

**Ans:**

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 = OPSS
```

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

2. Log in to the Oracle server as SYSTEM.

3. Create a user named OPSSuserid, 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);
```

1. 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"
```

[View All Answers](#)

### Question - 60:

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

**Ans:**

Currently, multibyte character sets are not supported for the WebLogic jDriver for Informix driver.

[View All Answers](#)

### Question - 61:

How is security handled in the WebLogic J2EE Connector Architecture?

**Ans:**

Due to the fact that the current configuration and packaging requirements for resource adapters in WebLogic Server require the hand-editing of the weblogic-ra.xml file, any new passwords specified in the security-principal-map entries are done in clear-text.

BEA understands the importance of protecting security passwords. Hence, we provide a Converter Tool that allows for the encryption of all passwords present in the weblogic-ra.xml file. The Converter Tool is shipped in the standard weblogic.jar file. For more information about the Password Converter Tool, see Using the Password Converter Tool in the Configuration section of the Programming the WebLogic J2EE Connector Architecture Guide at ../jconnector/index.html.

[View All Answers](#)

### Question - 62:

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

**Ans:**

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

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

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

[View All Answers](#)

### Question - 63:

While packaging the Web Application DefaultWebApp for deployment into the WebLogic server, the home and remote interfaces of the enterprise beans used by the servlets should reside in which directory?

**Ans:**

- a. DefaultWebApp/META-INF/classes
- b. DefaultWebApp/META-INF/lib
- c. DefaultWebApp/WEB-INF/lib
- d. DefaultWebApp/WEB-INF/classes
- e. DefaultWebApp/classes

Choice D is correct. When packaging a web application create META-INF and WEB-INF subdirectories in the application directory to hold deployment descriptors and compiled Java classes. All servlet classes and helper classes should reside in the WEB-INF/classes subdirectory. The home and remote interface classes for enterprise beans used by the servlets into the WEB-INF/classes subdirectory.

All the HTML files, JSP files, images, and any other files that these Web pages reference should exist in the application directory, maintaining the directory structure for referenced files. The META-INF directory contains the deployment descriptors for the enterprise beans, but not the classes.

[View All Answers](#)

### Question - 64:

Must EJBs be homogeneously deployed across a cluster? Why?

**Ans:**

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.

[View All Answers](#)

### Question - 65:

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

**Ans:**

Use the TxDataSource for bean-managed persistence.

[View All Answers](#)

### Question - 66:

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

**Ans:**

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.

[View All Answers](#)

### Question - 67:

The two primary cluster services provided by WebLogic Server are?

**Ans:**

- a. Http Session State Clustering
- b. File Service Clustering
- c. Time Service Clustering
- d. Object Clustering
- e. Event Clustering

Choices A and D are correct. A WebLogic Server cluster is a group of servers that work together to provide a more scalable and reliable application platform than a single server. A clustered service is an API or interface that is available on multiple servers in the cluster. HTTP session state clustering and object clustering are the two primary cluster services that WebLogic Server provides. WebLogic Server also provides cluster support for JMS destinations and JDBC connections. WebLogic Server provides clustering support for servlets and JSPs by replicating the HTTP session state of clients that access clustered servlets and JSPs. To benefit from HTTP session state clustering, you must ensure that the session state is persistent, either by configure in-memory replication, file system persistence, or JDBC persistence. If an object is clustered, instances of the object are deployed on all WebLogic Servers in the cluster. The client has a choice about which instance of the object to call. This is Object Clustering. The APIs and internal services that cannot be clustered in WebLogic Server version 6.0 are File services, Time services, WebLogic Events, Workspaces and ZAC.

[View All Answers](#)

### Question - 68:

I am 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.

**Ans:**

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.

[View All Answers](#)

### Question - 69:

Which of the following statements are true regarding MDBs (Message Driven Beans) on version 6.0 of WebLogic App Server?

**Ans:**

- a. MDBs support concurrent processing for both Topics and Queues.
- b. MDBs support concurrent processing for only Topics.
- c. MDBs support concurrent processing for only Queues.
- d. MDBs support concurrent processing neither Topics nor Queues.

Choice A is correct. MDBs support concurrent processing for both Topics and Queues. Previously, only concurrent processing for Queues was supported. To ensure concurrency, change the weblogic-ejb-jar.xml deployment descriptor max-beans-in-free-pool setting to >1. If this element is set to more than one, the container will spawn as many threads as specified. WebLogic Server maintains a free pool of EJBs for every stateless session bean and message driven bean class.

The max-beans-in-free-pool element defines the size of this pool. By default, max-beans-in-free-pool has no limit; the maximum number of beans in the free pool is limited only by the available memory.

[View All Answers](#)

### Question - 70:

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

**Ans:**

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.

[View All Answers](#)

### Question - 71:



What is error ORA-6502?

**Ans:**

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.

[View All Answers](#)

**Question - 72:**

How do I bind string values in a PreparedStatement?

**Ans:**

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();
```

[View All Answers](#)

**Question - 73:**

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

**Ans:**

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

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

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

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

[View All Answers](#)

**Question - 74:**

Which XML parser comes with WebLogic Server 6.1?

**Ans:**

We bundle a parser, based on Apache's Xerces 1.3.1 parser, in WebLogic Server 6.1. In addition, we include a WebLogic proprietary high-performance non-validating parser that you can use for small to medium sized XML documents. The WebLogic XML Registry allows you to configure the parser you want to use for specific document types.

[View All Answers](#)

**Question - 75:**

How do I restrict access to servlets and JSPs?

**Ans:**

The Java Servlet API Specification v2.2 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.

[View All Answers](#)

**Question - 76:**

Why ca not I boot WebLogic Server when using the LDAP Security Realm?

**Ans:**

When using an alternative security realm or a custom security realm in WebLogic Server, you must configure and enable the Caching realm.

When the LDAP server in Microsoft Site Server is installed and the root of the LDAP directory is created, a number of organizational units are created by default. Under Groups there is a default organization unit called NTGroups with a default group called Administrators which is empty. By default, WebLogic Server also provides a Group called Administrators that contains a member System which is the User under which WebLogic Server is started. If you use the defaults in Microsoft Site Server and start creating your own Groups under the default organizational units, WebLogic Server will not start. You need to create your own



organizational unit in the LDAP directory and create your Groups under that organizational unit.

If you have two Groups in the LDAP directory with the same name, WebLogic Server cannot properly authenticate the Users in the Groups. The LDAP security realm uses the Group's DN (distinguished name) to locate Groups in the LDAP directory. If you create more than one group with the same name, WebLogic Server only authenticates the Users in the first Group it locates. You must use unique Group names when using the LDAP security realm.

The LDAP realm V2 does not provide the following functionality provided in the LDAP realm V1:

- \* Listing all Users

- \* Listing the members of a Group

- \* The AuthProtocol and User Authentication mechanisms for authenticating the LDAP server.

The LDAP security realm needs to know where the Users and Groups are stored in the LDAP directory used with the security realm. This is done by specifying the distinguished names (DNs) of the LDAP directories containing the Users and Groups.

In LDAP, a DN starts with the leaf node and goes to the root node. The following figure shows a branch of a LDAP directory.

```
root
|
o=acme.com
|
ou=Groups
```

The DN for this branch would be specified as ou=Groups, o=acme.com.

In LDAP realm V1, you specify DN's via the GroupDN and UserDN attributes on the LDAPRealm MBean or through the Administration Console. However, you must reverse the DN's so the example DN would be specified as:

groupDN="o=acme.com, ou=Groups"

In LDAP realm V2, you specify DN's by adding user.dn and group.dn properties to the Configuration attribute of the CustomRealm MBean or through the Administration Console. Unlike LDAP realm V1, you do not have to reverse the DN. For example, the user.dn and group.dn properties for a LDAP realm V2 are specified as follows:

ConfigurationData="..., group.dn=ou=Groups, o=acme.com,..."

In short, the LDAP realm V1 requires reverse DN's and the LDAP realm V2 requires normal DN's.

A common error customers make when switching between LDAP realm V1 and LDAP realm V2 is copying over the reverse DN's thus causing the LDAP realm to stop working. Check your DN specifications when migrating from LDAP realm V1 to LDAP realm V2.

[View All Answers](#)

### Question - 77:

Which of the statements below is true for a web application using session management?

**Ans:**

- a.) The session object is invalidated, when the session times out.
- b.) The session object is invalidated, when sessionInvalidate() method of HttpSession is invoked.
- a. Both of the above statements are true.
- b. Only statement a.) is true.
- c. Only statement b.) is true.
- d. None of the above statements is true.

B is the correct choice. The session object will become invalid in either of the following scenarios:

- a.) When the session times out.
  - b.) When invalidate() method of the HttpSession interface is invoked.
- Please note that invalidate() and not sessionInvalidate() is the method of HttpSession interface. Thus choice B is correct.

[View All Answers](#)

### Question - 78:

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

**Ans:**

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
}
```

[View All Answers](#)

### Question - 79:

What is the function of T3 in WebLogic Server?

**Ans:**

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.

[View All Answers](#)

### Question - 80:



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

**Ans:**

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.

[View All Answers](#)

**Question - 81:**

How do I increase WebLogic Server memory?

**Ans:**

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.

[View All Answers](#)

**Question - 82:**

In CORBA, which of the following files generated by the idltojava compiler must be compiled before running the server application?

**Ans:**

- a. Only The Holder Classes
- b. Only The Helper and Holder Classes
- c. The implementation class(es) written by the user to provide body to the methods defined in the interface class
- d. All idltojava generated files (stubs, skeletons, helper, holder and interface classes)

Choice C and D are correct. The Helper class (a final class) provides auxiliary functionality, notably the narrow() method required to cast CORBA object references to their proper types. The Holder class provides operations for out and inout arguments, which CORBA has but which do not map easily to Java's semantics. The server application cannot be compiled successfully until the implementation class(es) has/have been written and compiled. All the other generated files however must be compiled to form '.class' files from the '.java' files. Thus all the implementation classes written by the user and all the "idltojava" generated files must be compiled to run the CORBA Server.

[View All Answers](#)

**Question - 83:**

The home of a Product CMP entity bean has a finder method, which returns an Enumeration of all the products whose price falls below a certain value, which is passed as the method argument. If there are no products in the database to match the above criteria what will be the result of a call to this finder method?

**Ans:**

- a. EJBException is thrown
- b. ObjectNotFoundException is thrown
- c. NoSuchEntityException is thrown
- d. An empty enumeration is returned

Choice D is correct. Find methods that return a single remote reference throw a FinderException if an application error occurs and a ObjectNotFoundException if a matching bean cannot be found. The ObjectNotFoundException is a subtype of FinderException and is only thrown by find methods, which return single remote references.

The findByPrimaryKey() method returns only one remote reference since there is a one-to-one relationship between a primary key's value and an entity. Find methods that return an Enumeration or Collection type return an empty collection or enumeration if no matching beans can be found or throw a FinderException if an application error occurs. NoSuchEntityException is a system exception thrown by the ejbLoad() method of an entity bean when the database row to be loaded is not found and also by the ejbStore() method when the database row to be updated cannot be found. It is a subclass of EJBException.

[View All Answers](#)

**Question - 84:**

Why do I get NoClassDefFound/Too Many Open files messages on Solaris?

**Ans:**

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.

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.

[View All Answers](#)

**Question - 85:**

How does WebLogic support CORBA and client communication via IIOP?

**Ans:**

"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.

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.

[View All Answers](#)

### Question - 86:

Which of the following are true about the transaction support in the WebLogic server?

**Ans:**

- a. WebLogic Server allows transactions to be terminated only by the client that created the transaction.
- b. WebLogic Server does not support multithreaded transactional clients.
- c. A client or a server object cannot invoke methods on an object that is participating in another transaction.
- d. WebLogic server supports Nested Transactions

Choices A and C are correct. WebLogic Server provides a Transaction Service that supports transactions in EJB and RMI applications. WebLogic Server allows transactions to be terminated only by the client that created the transaction. WebLogic Server implements the flat transaction model. Nested transactions are not supported.

WebLogic Server supports multithreaded transactional clients. Clients can make transaction requests concurrently in multiple threads. In WebLogic Server, a client or a server object cannot invoke methods on an object that is infected with (or participating in) another transaction. The method invocation issued by the client or the server will return an exception. Also in WebLogic Server, clients using third-party implementations of the Java Transaction API (for Java applications) are not supported.

[View All Answers](#)

### Question - 87:

Which of the following is NOT true about the security implementation in the WebLogic Server?

**Ans:**

- a. Servlets, JSPs, EJBs, RMI objects, and Java applications use the Java Authentication and Authorization Service to authenticate WebLogic Server.
- b. The default security realm in WebLogic Server is the File realm.
- c. The default authentication scheme for WebLogic Server is two-way authentication.
- d. An Administration Server may contain configuration information for one WebLogic Server or a cluster of WebLogic servers.

Choice C is correct because it is not true about the security in WebLogic server. A,B and D are true. Servlets, JSPs, EJBs, RMI objects, and Java applications use the Java Authentication and Authorization Service to authenticate WebLogic Server. JAAS is a standard extension to the Java 2 Software Development Kit. The authentication component of JAAS provides the ability to reliably and securely maintain client identity, regardless of whether the code is running as a Java application, a JSP, an EJB, an RMI object or a servlet.

In WebLogic Server, JAAS is layered over the existing Security Service Provider Interface (SPI) allowing the continued use of realm-based authorization. The default security realm in WebLogic Server is the File realm. When WebLogic Server is started, the File realm creates User, Group, and ACL objects from properties defined through the Administration Console in WebLogic Server and stored in the fileRealm.properties file. The File realm is designed for use with 1,000 or fewer users, for more no of users, an alternate security realm should be used. In WebLogic Server, an Administration Server is a WebLogic Server that functions as the central source of all configuration information. An Administration Server may contain configuration information for one WebLogic Server or a cluster of WebLogic servers.

[View All Answers](#)

### Question - 88:

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

**Ans:**

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

[View All Answers](#)

### Question - 89:

Can I configure the messaging bridge to automatically downgrade the quality of service if the Exactly-once service is not available on either the source or target bridge destination?

**Ans:**

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

[View All Answers](#)

### Question - 90:

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

**Ans:**

Yes, as long as the communication is between source and target WebLogic domains that are both running release 6.1 SP3 or later, and the bridge is configured to use the Exactly-once quality of service.

[View All Answers](#)

### Question - 91:

How do I use JTA transactions within an MDB?

**Ans:**

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();
}
}
```

[View All Answers](#)

### Question - 92:

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

#### Ans:

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

[View All Answers](#)

### Question - 93:

How does concurrency work for message-driven beans?

#### Ans:

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 JMS 6.1, 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 WLS 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).

[View All Answers](#)

### Question - 94:

How do I debug WebLogic Server using Visual Cafe 4.1?

#### Ans:

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:\beawlsrver6.1\configmydomain  
setEnv  
edit startWebLogic.cmd  
change "%JAVA\_HOME%\bin\java" -hotspot -ms64m -mx64m to visualcafeEEjdk13bindebugvm.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.

[View All Answers](#)

### Question - 95:

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

#### Ans:

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.

As of WLS 6.1 SP02 or later, you can use the Message Paging feature, which can free up valuable virtual memory during peak message load periods by swapping out messages from virtual memory to persistent storage when message loads reach a specified threshold.

[View All Answers](#)

### Question - 96:

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

#### Ans:

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.

[View All Answers](#)

### Question - 97:

What should an XPATH selector look like?

#### Ans:

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.

[View All Answers](#)

### Question - 98:

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

#### Ans:

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

[View All Answers](#)

### Question - 99:

A client invokes a method on a stateful session bean instance deployed in the WebLogic Server. While the method execution is in progress another method call arrives on the server. What will be the result?

#### Ans:

- a. RemoteException is thrown if the value of concurrency-strategy property is false
- b. EJBException is thrown if the value of concurrency-strategy property is false
- c. The EJB container blocks the concurrent method call and allows it to proceed when the previous call has completed if the value of allow-concurrent-calls is true
- d. In all cases, RemoteException is thrown

Choice C is correct. By default, simultaneous access to a stateful session EJB results in a RemoteException. However, you can set the allow-concurrent-calls option in the WebLogic EJB deployment descriptor to specify that a stateful session bean instance will allow concurrent method calls. This access restriction on stateful session EJBs applies whether the EJB client is remote or internal to WebLogic Server. By default, allows-concurrent-calls is false. However, when this value is set to true, the EJB container blocks the concurrent method call and allows it to proceed when the previous call has completed.

The concurrency-strategy element determines ejbLoad() and ejbStore() behavior for entity EJB instances.

[View All Answers](#)

### Question - 100:

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



### Ans:

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.

[View All Answers](#)

### Question - 101:

What are pending messages in the console?

### Ans:

Pending means the message could have been:

- \* sent in a transaction but not committed.
- \* received and not acknowledged.
- \* received and not committed.
- \* subject to a redelivery delay (as of WebLogic Server 6.1).
- \* subject to a delivery time (as of 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.

[View All Answers](#)

### Question - 102:

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

### Ans:

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);
    } 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.

[View All Answers](#)

### Question - 103:

How do I publish an XML message?

**Ans:**

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.

[View All Answers](#)

### Question - 104:

When should I use server session pools and connection consumers?

**Ans:**

WebLogic JMS implements an optional JMS facility for defining a server-managed pool of server sessions. This facility enables an application to process messages concurrently. A ConnectionConsumer object uses a server session to process received messages. If message traffic is heavy, the connection consumer can load each server session with multiple messages to minimize thread context switching. Multiple connection consumers can share server sessions in a server session pool. To learn how to use the connection consumers within an application, see the section Processing Messages Concurrently in Programming WebLogic JMS, or the javax.jms.ConnectionConsumer javadoc.

Note: Server session pools can also be implemented using Message Driven Beans. Using MDBs is preferable to using server session pools - see the answer to the question, "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.

[View All Answers](#)

### Question - 105:

What is the NO\_ACKNOWLEDGE acknowledge mode used for?

**Ans:**

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. v 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.

[View All Answers](#)

### Question - 106:

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

**Ans:**

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`).

[View All Answers](#)

### Question - 107:

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

**Ans:**

Ways to get a thread dump:

\* Try running this from the command line (after running the `setEnv` script in `/bea/wlserver6.1/config/mydomain/`):

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

\* On Windows, from the console window, enter `Ctrl+Break`.

\* On UNIX, signal the server using `kill -3`.

[View All Answers](#)

### Question - 108:

The `MaxPostTimeSecs` attribute set in the Administration console under Servers or virtual hosts section corresponds to which of the following?

**Ans:**

a. The amount of time that WebLogic Server waits between receiving chunks of data in an HTTP POST.

b. The total amount of time that WebLogic Server spends receiving HTTP POST data.

c. The time spent by WebLogic server to post data to other servers in the cluster.

d. The number of bytes of data received in a POST from a single request.

Choice B is correct. Web servers may face denial-of-service attacks, which is usually carried out by sending huge amounts of data in an HTTP POST method. You can set three attributes in WebLogic Server that help prevent this type of attack. These attributes are set in the console, under Servers or virtual hosts. You can limit the amount of time that WebLogic Server waits between receiving chunks of data in an HTTP POST by setting the attribute `PostTimeoutSecs`.

The `MaxPostTimeSecs` attribute limits the total amount of time that WebLogic Server spends receiving post data. If this limit is triggered, a `PostTimeoutException` is thrown and a message is sent to the server log. `MaxPostSize` attribute limits the number of bytes of data received in a POST from a single request. If this limit is triggered, a `MaxPostSizeExceeded` exception is thrown and a message is sent to the server log.

[View All Answers](#)

### Question - 109:

How should I use sorted queues?

**Ans:**

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.

[View All Answers](#)

### Question - 110:

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

**Ans:**

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.

[View All Answers](#)

### Question - 111:

In the WebLogic server, if stateless session bean instances are getting frequently created and removed, performance can be improved by setting a high value for which of the following?

**Ans:**

a. `max-beans-in-free-pool`

b. `max-beans-in-cache`

c. `max-beans-in-memory`

d. `max-stateless-beans-in-cache`

Choice A is correct. WebLogic Server maintains a free pool of EJBs for every stateless session bean class. The `max-beans-in-free-pool` element defines the size of this pool. By default, `max-beans-in-free-pool` has no limit; the maximum number of beans in the free pool is limited only by the available memory.

When EJBs are created, the session bean instance is created and given an identity. When the client removes a bean, the bean instance is placed in the free pool. When you create a subsequent bean, you can avoid object allocation by reusing the previous instance that is in the free pool. So the `max-beans-in-free-pool` element can improve performance if EJBs are frequently created and removed. Keeping this parameter too high uses extra memory and keeping it too low causes unnecessary



object creation.

WebLogic Server allows you to configure the number of active beans that are present in the EJB cache (the in-memory space where beans exist). The max-beans-in-cache element specifies the maximum number of objects of this class that are allowed in memory. When max-bean-in-cache is reached, WebLogic Server passivates some EJBs that have not been recently used by a client. Choices C and D are not valid properties.

[View All Answers](#)

### Question - 112:

Which of the following statements are true regarding the identity of two EJBs?

**Ans:**

- a. Two stateful session beans are identical if their data attributes are identical.
- b. Two stateful session beans are identical if their session contexts are equal.
- c. Two stateless session beans are identical if they are of the same type.
- d. Two stateless session beans are identical if their session contexts are equal.
- e. Two entity beans are identical if they have same primary key but different home interface.
- f. Two entity beans are identical if they have different primary key but same home interface.

B and C are correct. Since the stateful session beans maintain the conversational state of the clients, they are identical when their session contexts are equal. Two stateful session beans may have identical data attributes, but if the session contexts are different they are not identical. Thus choice A is incorrect and B is correct. Since stateless beans do not retain the conversational state, they are considered identical if they are of the same type. Thus choice C is correct.

If two entity objects have the same home interface and primary key, they are considered identical. The EJB specification does not mention object equality based on the `=` operator. Also, if you compare two object references using the Java API, `Object.equals(Object obj)`, the result is unspecified. The only way to compare object equality is through the `isIdentical(EJBObject)` API. Thus choice E and F are incorrect.

[View All Answers](#)

### Question - 113:

Does WebLogic JMS support clustering?

**Ans:**

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 MDBs 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 MDBs 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 EJBs in general that try to reference a non-local JMS destination.

[View All Answers](#)

### Question - 114:

How do the WLS JMS 6.1 server/destination message maximum and threshold values work?

**Ans:**

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.

[View All Answers](#)

### Question - 115:

How do I use persistence?

**Ans:**

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 overwritten, as it is a vendor-only method.

3. Make sure you are calling either:

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

```
-- or --
```

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

```
-- or --
```

set the `DefaultDeliveryMode` mode on connection factory in the config.xml file to persistent (the `QueueSender.setDeliveryMode` 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 8.1.7 or later is needed to run with WLS 6.1 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) {
}
}
}
```

[View All Answers](#)





### Question - 116:

The Multicast TTL setting for a cluster in the WebLogic Admin console sets which of the following values?

#### Ans:

The Multicast TTL setting for a cluster in the WebLogic Admin console sets which of the following values?

- a. Maximum time taken for multicast messages to reach their final destination
- b. The number of routers a multicast message can pass through before the packet can be discarded
- c. The multicast address to be used by the messages sent from the cluster
- d. Minimum time taken for broadcasting a multicast message from the cluster

Choice B is correct. The Multicast TTL (TTL-Time to Live) setting specifies the number of routers a multicast message can pass through before the packet can be discarded. To configure the multicast TTL for a cluster, you should change the Multicast TTL value in the WebLogic Server administration console. This sets the number of network hops a multicast message makes before the packet can be discarded.

If you choose to distribute a cluster over a WAN (or across multiple subnets), you must plan and configure your network topology to ensure that multicast messages are reliably transmitted to all servers in the cluster. One of the requirements to be met by the network is that the multicast Time To Live (TTL) value must be high enough to ensure that routers do not discard multicast packets before they reach their final destination.

[View All Answers](#)

### Question - 117:

Which types of JDBC databases does WebLogic JMS support?

#### Ans:

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)

[View All Answers](#)

### Question - 118:

How do I use a temporary destination?

#### Ans:

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 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.

[View All Answers](#)

### Question - 119:

Why does JMS Session.createTopic or JMS Session.createQueue fail to create a destination in WLS JMS 6.1 (it worked in 5.1)?

#### Ans:

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 subsection 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 (JMSEException e) {
    e.printStackTrace();
}
}
```

[View All Answers](#)

### Question - 120:

How do I configure JMS security?

#### Ans:

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.topic.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.

[View All Answers](#)

### Question - 121:

Are there C/C++ interfaces to WLS JMS?

#### Ans:

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.

[View All Answers](#)

### Question - 122:

Why am I getting an ORA-01000: maximum open cursors exceeded error, even though I closed all ResultSet, Statement, and Connection objects?

#### Ans:

This is an Oracle issue. According to Oracle's documentation, dynamic cursors can remain open from run to run in a session and are not closeable when a procedure closes. To work around this issue, you can increase the number of open cursors allowed in the database or you can reset the connection pool (close and reopen database connections in the connection pool).

To reset the connection pool, you can untarget and retarget the connection pool using the Administration Console. You can also use the reset() method through the JMX API or the RESET\_POOL command on the WebLogic Server command line interface.

[View All Answers](#)

### Question - 123:

How do I prevent errors when running t3dbping?

#### Ans:

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.

[View All Answers](#)

### Question - 124:

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

**Ans:**

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.

[View All Answers](#)

### Question - 125:

How do I learn what codesets are available in Oracle?

**Ans:**

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='AL24UTFFSS';
This would produce the following response if the codeset is installed:
VALUE
-----
AL24UTFFSS
```

You can use Oracle's installation tools to install additional codesets. Contact Oracle for more information.

[View All Answers](#)

### Question - 126:

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

**Ans:**

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.

[View All Answers](#)

### Question - 127:

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

**Ans:**

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.

[View All Answers](#)



### Question - 128:

What causes an OCIW32.dll error?

#### Ans:

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.

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

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.jdbctools.v2\_0.LoginFailureException:

TRANSACTION\_READ\_UNCOMMITTED isolation level not allowed

The Stack Trace:

com.roguewave.jdbctools.v2\_0.LoginFailureException:

TRANSACTION\_READ\_UNCOMMITTED isolation level not allowed

at

com.roguewave.jdbctools.v2\_0.jdbc.JDBCServer.createConnection

(JDBCServer.java :46)

at com.roguewave.jdbctools.v2\_0.ConnectionPool.getConnection\_

(ConnectionPool.jav a:412)

at com.roguewave.jdbctools.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.

[View All Answers](#)

### Question - 129:

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

#### Ans:

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

server=machineName

port=instancePort

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

[View All Answers](#)

### Question - 130:

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

#### Ans:

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

[View All Answers](#)

### Question - 131:

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

#### Ans:

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

java.rmi.UnmarshalException: error unmarshalling arguments; nested

exception

is:

java.lang.ClassNotFoundException:

com.mycompany.InteractionSpecImpl

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

[View All Answers](#)

### Question - 132:

How do I set up my CLASSPATH?

#### Ans:

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. 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.



[View All Answers](#)

### Question - 133:

Which of the following are recommended practices to be performed in the `ejbPassivate()` method of a stateful session bean?

**Ans:**

- a. Close any open resources, like database connections
- b. All non-transient, non-serializable fields(except some special types) should be set to null.
- c. All transient fields should be set to null
- d. Make all database connection reference fields transient
- e. All primitive type fields should be set to null

Choices A, B and D are correct. When a bean is about to be passivated, its `ejbPassivate()` method is invoked, alerting the bean instance that it is about to enter the Passivated state. At this time, the bean instance should close any open resources and set all non transient, non serializable fields to null. This will prevent problems from occurring when the bean is serialized. Transient fields will simply be ignored. Serializable fields will be saved. Open resources such as sockets or JDBC connections must be closed whenever the bean is passivated. In stateful session beans, open resources will not be maintained for the life of the bean instance. When a stateful session bean is passivated, any open resource can cause problems with the activation mechanism.

A bean's conversational state may consist of only primitive values, objects that are serializable, and the following special types-SessionContext, EJBHome, EJBObject, UserTransaction and Context (only when it references the JNDI ENC) . The types in this list (and their subtypes) are handled specially by the passivation mechanism. They don't need to be serializable; they will be maintained through passivation and restored automatically to the bean instance when it is activated

[View All Answers](#)

### Question - 134:

Why did my JDBC code throw a rollback SQLException?

**Ans:**

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.

[View All Answers](#)

### Question - 135:

How do stubs work in a WebLogic Server cluster?

**Ans:**

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.

[View All Answers](#)

### Question - 136:

I tried to run two of the applets in the examples directory of the distribution?

**Ans:**

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.

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.

[View All Answers](#)

### Question - 137:

Can I use a native two-tier driver for a browser applet?

**Ans:**

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 `jdbcDriver` for Oracle (or any other two-tier JDBC driver) for the connection between the WebLogic Server and the DBMS.

[View All Answers](#)

### Question - 138:

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

**Ans:**



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.

[View All Answers](#)

### **Question - 139:**

What is BEA Weblogic?

#### **Ans:**

BEA WebLogic is a J2EE application server and also an HTTP web server by BEA Systems of San Jose, California, for Unix, Linux, Microsoft Windows, and other platforms. WebLogic supports Oracle, DB2, Microsoft SQL Server, and other JDBC-compliant databases. WebLogic Server supports WS-Security and is compliant with J2EE 1.3.

BEA WebLogic Server is part of the BEA WebLogic Platform. The other parts of WebLogic Platform are:

- \* Portal, which includes Commerce Server and Personalization Server (which is built on a BEA-produced Rete rules engine),
- \* WebLogic Integration,
- \* WebLogic Workshop, an IDE for Java, and
- \* JRockit, a JVM for Intel CPUs.

WebLogic Server includes .NET interoperability and supports the following native integration capabilities:

- \* Native enterprise-grade JMS messaging
- \* J2EE Connector Architecture
- \* WebLogic/Tuxedo Connector
- \* COM+ Connectivity
- \* CORBA connectivity
- \* IBM WebSphere MQ connectivity

BEA WebLogic Server Process Edition also includes Business Process Management and Data Mapping functionality.

WebLogic supports security policies managed by Security Administrators. The BEA WebLogic Server Security Model includes:

- \* Separate application business logic from security code
- \* Complete scope of security coverage for all J2EE and non-J2EE components

[View All Answers](#)

# Applications Programs Most Popular & Related Interview Guides

- 1 : [AutoCAD Interview Questions and Answers.](#)
- 2 : [Microsoft Office Interview Questions and Answers.](#)
- 3 : [Microsoft Outlook Interview Questions and Answers.](#)
- 4 : [Microsoft Excel Interview Questions and Answers.](#)
- 5 : [MATLAB Interview Questions and Answers.](#)
- 6 : [WPF Interview Questions and Answers.](#)
- 7 : [OOAD Interview Questions and Answers.](#)
- 8 : [Quickbook Interview Questions and Answers.](#)
- 9 : [Microsoft Word Interview Questions and Answers.](#)
- 10 : [Bioinformatics Interview Questions and Answers.](#)

Follow us on FaceBook

[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/  
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)