BEAWebLogic Server®

Known and Resolved Issues

Contents

WebLogic Server Known and Resolved Issues

WebLogic Server Known and Resolved Issues

Note: Because this information documents a Tech Preview of a BEA product, it may be incomplete and is subject to change.

The following sections describe known problems in WebLogic Server 10.3, as well as problems that were resolved in earlier versions. Entries include a description of the problem, and a workaround or solution where appropriate. A notation in the Fixed In column indicates that the problem has been resolved.

For information about new and changed functionality in WebLogic Server 10.3 Tech Preview, see Introduction to WebLogic Server 10.3 Tech Preview.

- FastSwap Deployment

- JDBC Support

- Building Web Services Using Ant

FastSwap Deployment

JPA and JDO persistent classes are not eligible for redefinition via FastSwap. Additionally, the FastSwap version in the WebLogic Server Tech Preview may cause errors if your application uses runtime-enhanced, non-annotated persistent types. Only enable FastSwap if your persistent classes are pre-enhanced, or if they use the `@Entity/@Embeddable/@MappedSuperclass` annotations.

JDBC Support

In order to support the new JDBC 4.0 features in the WebLogic Server Tech Preview, WebLogic Server must be used with a JDBC 4.0 driver . Currently, WebLogic Server does not include any JDBC 4.0 drivers. To use the JDBC 4.0 features, customer must obtain a third-party JDBC 4.0 driver from the driver vendor and install the driver. For an example, see http://e-docs.bea.com/wls/docs100/jdbc_admin/third_party_drivers.html.

WebLogic Server supports JDBC 4.0 in the following manner:

- SQLXML support is limited to server -side. RMI clients do not support SQLXML.

- Calls to Wrapper.unwrap() from RMI clients throws SQLException (Unsupported feature).

- The Statement poolable application programming interface (API) has the following limitations:

  - Call `Statement isPoolable()` always returns false

  - Call `Statement.setPoolable()` is no op.

  - Call `PreparedStatement.setPoolable(false)` removes the prepared statement or callable statement from the statement cache

WebLogic Server includes two datasource module files:
`com.bea.core.datasource6_1.3.0.0.jar` and
`com.bea.core.datasource_1.3.0.0.jar`.

You can use either of these two files in a client, but not both at the same time. See the following guidelines:

- If you need to use JDBC 4.0 new features, use `com.bea.core.datasource6_1.3.0.0.jar` with JDK 6 only.

- If using JDBC 4.0 new features is not a consideration, you can use either module file. However, the `com.bea.core.datasource6_1.3.0.0.jar` can only be used with JDK 6.

Building Web Services Using Ant

Due to a bug with the classloader in the JDK 6 that is installed with the product, when building your Web Services using Ant, you may receive the following error:

```
java.lang.NoSuchMethodException
```

To workaround this issue, you must update your Ant script to include the
$JAVA_HOME/lib/tools.jar file in the CLASSPATH.

For example:

```
:runAnt
if "%_USE_CLASSPATH%"=="no" goto runAntNoClasspath
if not "%CLASSPATH%"=="" goto runAntWithClasspath
"%_JAVACMD%" %ANT_OPTS% -classpath
"%ANT_HOME%\lib\ant-launcher.jar;c:\bea\jdk160_02\lib\tools.jar"
"-Dant.home=%ANT_HOME%" org.apache.tools.ant.launch.Launcher %ANT_ARGS%
%ANT_CMD_LINE_ARGS%
goto end
:runAntNoClasspath
"%_JAVACMD%" %ANT_OPTS% -classpath
"%ANT_HOME%\lib\ant-launcher.jar;c:\bea\jdk160_02\lib\tools.jar"
"-Dant.home=%ANT_HOME%" org.apache.tools.ant.launch.Launcher %ANT_ARGS%
%ANT_CMD_LINE_ARGS%
goto end
:runAntWithClasspath
"%_JAVACMD%" %ANT_OPTS% -classpath "%ANT_HOME%\lib\ant-
launcher.jar;c:\bea\jdk160_02\lib\tools.jar"
"-Dant.home=%ANT_HOME%" org.apache.tools.ant.launch.Launcher %ANT_ARGS%
-cp "%CLASSPATH%" %ANT_CMD_LINE_ARGS%
goto end
```

This issue will be fixed in a future release of the JDK 6. Sun JDK 1.6.0 users should also review
the following document: Using JDK6 With Web Services.