BEA WebLogic Workshop Release Notes

BEA WebLogic Workshop Release 7.0
Date: June 2002

This document includes the following topics:

For updated release note information, go to the BEA documentation Web site at the following URL:

http://edocs.bea.com

Platform Support and System Requirements

For information on platform support, including hardware and software requirements, see the Supported Platforms page at the following location:

http://edocs.bea.com/platform/docs70/support/index.html

Migrating Web Services to the Released Version

If you are upgrading from a beta build of WebLogic Workshop, you should be aware of the following changes from the beta version to the released version. In some cases, you may need to upgrade application code.

Migrating from Beta 1

If you are upgrading directly from the Beta 1 (build number 2002.0221.1) release of WebLogic Workshop. These changes are not needed when upgrading from Beta 2 (build number 2002.0423.3). You can find which version you are running by clicking Help, then clicking About WebLogic Workshop.

- All ECMAScript files must reflect the new file extension (.jsx instead of .js).
- All JSX files must reflect the new import syntax: The `importClass "[package.]class";` statement in JSX file has changed to `import [package.]class;`

- You must regenerate WSDL files and CTRL files for Service controls and EJB controls you created with the beta version. This is due to syntax and namespace changes that have rendered the generated portions of the beta CTRL files no longer accurate.

  Note that if you have modified the generated CTRL file (such as by adding XML maps to it), you must reapply the changes to the newly generated control in order to use them.

  For more information on generating controls, see How Do I: Generate a CTRL File? For more information on generating WSDL files, see How Do I: Generate a WSDL File?

- JAR files for EJBs used by your application should now be placed in the BEA_HOME\weblogic700\samples\workshop\applications directory. (Previously, they were in the \samples\workshop directory.)

Migrating from Beta 2

- Client proxy code is now stored in multiple classes, in a single JAR file. Prior to release, the client proxy code generated by WebLogic Workshop was stored in a single class. In the released version the proxy code is stored in multiple classes in a single JAR file. You can reach the client proxy code from the Overview tab in Test View. For more information on using client Java proxies, see How Do I: Use the Java Proxy for a Web Service?
- Delete the .Workshop file before installing the BEA WebLogic Platform. If you have previously used a beta version of WebLogic Workshop, you should delete the .Workshop file installed with WebLogic Workshop before installing. The .Workshop file is located in your home directory. On Windows systems, the home directory is indicated by the %USERPROFILE% environment variable. On UNIX and Linux systems, the .Workshop file is located in the ~ (tilde) directory (sometimes also represented in the environment by $HOME). For more information about the .Workshop file, see .Workshop Configuration File in the WebLogic Workshop documentation.

Web Services Built with WebLogic Workshop Support Source Compatibility, Not Binary Compatibility

Compiled web services built with a beta version of WebLogic Workshop must be recompiled to run on the final version of WebLogic Workshop.

Setting the Maximum Size of Conversation IDs

Conversational web services use a database to store conversation IDs - the unique strings that identify conversations. The database table used to store conversational state is created if it doesn't exist upon the first use of a service. The maximum size for

conversation IDs is built into the database schema. The default maximum size allowed for conversation IDs is 768 characters. Ordinarily, 768 works well as a maximum. However, if the design of a particular service requires a different maximum length, you can change it in two ways as described in this note.

First, you must set the weblogic.jws.ConversationMaxKeyLength property. This property can be set in either of the two following ways. Both of these require that you restart the server after making the change:

Edit the startWebLogic command file

1. Locate one of the following files, depending on which operating system you are using: startWebLogic.cmd (on Windows) or startWebLogic.sh (on UNIX or Linux).

   This file is located in the domain directory of your project. For example, for the samples project installed with WebLogic Workshop, this is the samples\workshop directory.

2. Locate the following text in the file:

- On Windows, find the line with the following text: `@set JAVA_OPTIONS=%JAVA_OPTIONS% %JAVA_PROPERTIES%`
- On Linux or UNIX, find the first line with the following text: `JAVA_PROPERTIES=`

3. Append the following to the line, replacing <numberOfCharacters> with the new maximum length:

   -Dweblogic.jws.ConversationMaxKeyLength=<numberOfCharacters>

Edit the jws-config.properties file

1. Locate the jws-config.properties file.

   This file is located in the domain directory of your project. For example, for the samples project installed with WebLogic Workshop, this is the samples\workshop directory.

2. Change the value of the weblogic.jws.ConversationMaxKeyLength property. The syntax is as follows:

   weblogic.jws.ConversationMaxKeyLength=<numberOfCharacters>

After restarting the server, you must use your service (for example, by calling one of its methods) so that the conversational state table will be recreated with the new maximum conversation ID length. In the event that the conversational state tables already exist, you

will need to drop them prior to referencing the service so that they will be recreated. After dropping the tables, and restarting the server, the first reference to your service will cause the tables to be recreated with the new value for the maximum conversation ID length.

Note that a different table is used for each service.

Known Limitations

Working with Databases

Out of Memory Error for Large Result Sets

You may receive an "Out of memory" error when retrieving a large ResultSet using a Database control. This is because the query result is converted to HTML, which more than doubles its size.

Workaround: Try to use queries that return smaller ResultSets or return an Iterator.

"Invalid Transaction State" When Using Pointbase

If you are running Weblogic Workshop with the Pointbase database (the default configuration for Workshop), and you encounter an error stating that the database is in an "invalid transaction state", your pointbase database files have become corrupted and you must replace them.

Workaround: Replace cajun.dbn and cajun$1.wal, two Pointbase files included with Weblogic Workshop samples. These are located in the WebLogic Workshop samples domain directory. Usually this directory is at /bea/weblogic700/samples/workshop. If you have made back up copies of these two files, you can replace the two files in use with your clean back ups.

You can also download clean copies of the files from the bug fixes section of the resource library at dev2dev Online. On the bug fixes page, look for change request (CR) number CR080632, then click the name of the bug to reach a page from which you can download the files. Note that any information you had stored in your Pointbase database will be lost if you replace the files with the clean files downloaded from the web.

Testing and Debugging

Problems When Using Netscape 6.2

- If you test a JWS file and your browser is Netscape 6.2, the browser may hang in the splash screen when the browser is launched (as when you select "Start" or

"Start and Debug" from WebLogic Workshop). This is a bug in Netscape 6.2 that is tracked in Bugzilla bug numbers 54701 and 54716.
- If you test a JWS file in a Netscape 6.2 browser window and try to use the Callback WSDL link on the Overview page, you will get a blank page instead of a generated WSDL.

Workaround: Use a different browser.

WebLogic Server Does Not Respond While Debugging Multiple Service Instances

When debugging a web service using the breakpoints and the Step Into command, WebLogic Server can reach a state in which it no longer responds to commands. This can happen if you run multiple instances of your web service with breakpoints set. For example, you might inadvertently double-click the Test View button corresponding to a method of your service, then use the Step Into command twice (through the Debug menu or toolbar button).

Workaround: If you reach a state in which your project will no longer build, stop and restart WebLogic Server.

Execution Sometimes Moves to Unexpected Locations When Stepping Through Code

In the following situations, you might find that the debugger takes execution to unexpected lines in your code:

- When using the Step Out command, execution moves to the first line of the calling procedure, rather than the line following the call.
- When stepping through a `try`/`catch`/`finally` block, execution moves to the `try` statement after the `catch` and `finally` statements finish executing.
- When stepping through inner class constructors, execution moves to the declaration of the web service class. Doing this repeatedly may cause the debugger to stop responding.

Delay When Attempting to Examine Values While Debugging

You might experience delay when you try to examine variable values while debugging code with large objects. In these cases, WebLogic Workshop will display ... or no value until the value appears.

Cannot Step into Service Control Instance

When debugging, you can't step into a Service control instance. This version of WebLogic Workshop does not allow you to "step into" on a call to a Service control method.

Workaround: If the called web service's source code (JWS file) is available, you can set a breakpoint there.

Test View Does Nothing When Testing with a Large XML Document

When you have a parameter map for a method, the instance of the XML document you enter in the text area for a parameter can exceed the maximum allowable HTTP GET string. When this occurs the browser will appear to accept the button press but will do nothing.

Workaround: Submit a smaller test document or use the Test XML page instead of the Test Form page.

WebLogic Workshop on RedHat Linux

Required Environment Setting on Linux 7.1 and Earlier

In order to use WebLogic Workshop on on versions of RedHat Linux prior to 7.2, you must include the following among your environment settings:

export LD_ASSUME_KERNEL=2.2.5

Asynchrony and Callbacks

Client Callbacks Can Fail Silently

When a callback is invoked on a client that has not provided a callback URL, a RuntimeException is generated but no error is reported automatically in Test View or in workshop.log. You are not required to handle RuntimeExceptions, so the error is completely silent by default. If you want to catch occurrences of this situation, you must implement a handler for the `JwsContext.onException` callback in your JWS file. For more information on the callback, see How Do I: Handle Errors In a Web Service? in the WebLogic Workshop documentation.

Client Proxy Code

XML Maps Must Be Namespace-Qualified In Order to Produce Valid Client Proxy JAR Files

If your web service has XML maps that do not have namespace declarations, the client proxy JAR file generated for that service will be empty. This is due to constraints of the underlying proxy generation code.

Workaround: Add a namespace declaration to the root element of the XML maps, as with the red text in the following example:

```
/**
* @jws:operation
* @jws:parameter-xml xml-map::
* <person xmlns="http://www.openuri.org/">
* <lastname><xm:value obj="String lastName"/></lastname>
* <firstname>{firstName}</firstname>
* </person>
* ::
*
* @return the string "Received person: '{name}'"
*/
public String acceptPerson(String lastName, String firstName)
```

Deployment

WebLogic Workshop Uses a Single JMS Server for Certain Messaging Needs

WebLogic Workshop applications can easily be deployed on WebLogic Server cluster environments using the instructions provided in the WebLogic Workshop documentation in Clustering Workshop Web Services.

In version 1 of WebLogic Workshop, however, applications that use a JMS message queue rely on the services of a single JMS server and connection factory. This includes applications that use JMS as a transport protocol, the `message-buffer` property, or Timer controls. Future versions of WebLogic Workshop will use the distributed JMS destination capability in WebLogic Server to reduce the dependency on a single JMS server in a clustered environment.

"Unexpected Exception" When Deploying Secure Web Service

When developing a secure web service, you may see the following error message:

An unexpected exception occurred while attempting to deploy the Enterprise JavaBean for this Web Service.

This can happen if there are syntax errors in the web.xml or weblogic.xml files edited to include security information. To confirm that this is the cause of the error message, look in the workshop.log file, which contains exception information logged by a web service. There, you may find specific information about configuration errors.

The workshop.log file is located in the root of the domain in which your web service is deployed. For example, for samples installed with WebLogic Workshop, this is the samples\workshop folder.

Callback Interface Must Be Public on Production Server

Callback interfaces must be declared as "public" in production mode. If you don't declare callback interfaces as public, dynamic proxy generation fails with "IllegalAccessException" in production mode.

UDDI Explorer

UDDI Overview Page Not Rendered in Mozilla on Linux

On Linux, when using the UDDI Explorer to browse for web services, the Mozilla web browser may not render the content of the service's overview page. Note that the page's source code is there, but does not render in Mozilla.

Documentation

Use the "AND" Operator to Narrow Search Results

When using the Search tab in help for WebLogic Workshop, be sure to join search terms with AND when you want to return topics that include all of the search terms. By default the search engine uses OR for multiple search terms separated by spaces. This means that without AND, a search on the following string will return any topic containing one or more of the terms:

web service database access

To narrow this search to include only those topics that include *all* of the search terms, enter the following instead:

web AND service AND database AND access

Netscape 4.76 May Not Launch from WebLogic Workshop

If your browser is Netscape 4.76, launching Help from WebLogic Workshop may not work. This is a bug in Netscape 4.76 related to launching the browser from another application. Please upgrade to at least Netscape 4.78.