



BEA Workshop for WebLogic Platform™

Release Notes

Version 9.2 Maintenance Pack 3
Revised: March 2008

Contents

- What's New in BEA Workshop for WebLogic Platform 1
 - Built on Eclipse 3.1. 1
 - Apache Beehive 2
 - Java Server Faces 2
 - Standards-based Java Web Services 2
 - Java 5 Annotations 3
 - Upgrade Tools. 3
 - Blended Application Support 3
- Considerations for Using Workshop for WebLogic Version 9.2.3 3
- Location of Supported Platform Information. 4
- Documentation Updates. 4
- Known Limitations for Workshop for WebLogic Version 9.2.3. 4
- Resolved Problems for Workshop for WebLogic Version 9.2.3. 13
- Resolved Problems for Workshop for WebLogic Version 9.2.2. 14
- Resolved Problems for Workshop for WebLogic Version 9.2.1 16

BEA Workshop for WebLogic Platform Version 9.2.3 Release Notes

This document contains information on the following subjects:

- [What's New in BEA Workshop for WebLogic Platform](#)
- [Considerations for Using Workshop for WebLogic Version 9.2.3](#)
- [Location of Supported Platform Information](#)
- [Documentation Updates](#)
- [Known Limitations for Workshop for WebLogic Version 9.2.3](#)
- [Resolved Problems for Workshop for WebLogic Version 9.2.3](#)
- [Resolved Problems for Workshop for WebLogic Version 9.2.2](#)
- [Resolved Problems for Workshop for WebLogic Version 9.2.1](#)

What's New in BEA Workshop for WebLogic Platform

Workshop for WebLogic continues the ground-breaking innovation of the 8.1 release, providing powerful tools for developing WebLogic Platform applications.

Here are some of the most important new features in Workshop for WebLogic:

Built on Eclipse 3.1

Workshop for WebLogic is built on the widely used Eclipse Platform. Instead of the proprietary IDE framework used in previous releases, version 9.2.3 uses the Web Tools Platform 1.0.

Apache Beehive

Version 9.2.3 supports Apache Beehive, an open source framework for web applications. Support for Apache Beehive includes:

- **Beehive NetUI:** A web application framework based on Apache Struts. Introduced in version 8.1 as Page Flow technology, Beehive NetUI has evolved into an open-source web application framework. Version 9.2.3 includes advanced tooling features to help developers build Beehive NetUI applications, including graphical tools for JSP and Page Flow controller authoring.
- **Beehive Controls:** A lightweight component framework based upon annotated JavaBeans. Beehive controls provide an easy to use client model for accessing a variety of J2EE resource types. The built-in controls introduced in version 8.1 (database, EJB, JMS and web service controls) are available as system controls in version 9.2.3. You can also create your own custom controls using a new, transparent, POJO architecture.

Java Server Faces

Version 9.2.3 includes first-class support for Java Server Faces technology, including:

- Support for JSF pages in NetUI Beehive applications
- Tools support for JSF backing files and event handlers
- Integration of JSF with the Beehive NetUI life cycle

Standards-based Java Web Services

The same ground-breaking web service support in version 8.1 has been carried forward in version 9.2.3, now built on JSR-181. Web service support includes:

- Asynchronous callbacks
- State management with conversations
- Ease of iterative development with the test browser has been carried forward from version 8.1
- Reliable messaging with JMS and WS-Reliability
- Digital Signature and HTTPS support with WS-Security/WS-Policy

Java 5 Annotations

As with version 8.1, version 9.2.3 supports the use of annotations to simplify the development of complex components. While most of the functionality of the version 8.1 annotations carries forward into this version, the version 9.2.3 annotations are based on the JSR-175 standard, new in Java 5. Workshop for WebLogic continues to provide tool support to keep the use of annotations simple, including a property editor for intuitive annotation editing.

Upgrade Tools

Version 9.2.3 makes it easy to upgrade your version 8.1 applications. Upgrade support includes:

- Pre-upgrade reporting that doesn't affect the original source code
- Automatic handling of application and project structure upgrade
- Automatic file extension renaming
- Wizard and command line versions of the upgrade tool are available

Blended Application Support

Workshop for WebLogic supports blended application architecture combining the best of open source and BEA's innovative technologies. Blended application offers:

- Long-term, license-independent support for applications
- Best of breed hybridization
- Transparency and predictability

Considerations for Using Workshop for WebLogic Version 9.2.3

Workshop for WebLogic is targeted toward the iterative development experience rather than production deployment. As such, a number of features that work correctly in a standalone (development) server environment will not function as expected in a clustered deployment.

Important — Development and testing of applications using this release of Workshop for WebLogic should be done using standalone server environments.

Location of Supported Platform Information

For more information on platform support, including hardware and software requirements, see the [Supported Platforms](#) web site.

Documentation Updates

Updated documentation is available at the [Workshop for WebLogic e-docs site](#).

Known Limitations for Workshop for WebLogic Version 9.2.3

[Table 1](#) lists the known limitations found in Workshop for WebLogic.

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR242934	On upgrade failure, files are not reverted to their original state When an upgrade fails the new files are not reverted to their original state but may not be fully upgraded. For example, the file extension may have been changed to .java, but annotations may not have been translated to Workshop for WebLogic version 9.2.1 style. Note that the original source files are not altered. Platform: All Workaround: Either drag and drop the version 8.1 file into the workspace and perform upgrade on that file again or re-import the entire project.
CR249614	Web service with modified parameter names requires @WebParam annotation post-upgrade A start-from-WSDL web service where the parameter names (in the Java signature) do not match the WSDL name require a @WebParam annotation be manually added post-upgrade. The most likely case where a parameter name is modified is when the WSDL name is an invalid Java identifier. Platform: All Workaround: Add a @WebParam annotation with the matching WSDL name to each parameter in the web service operation.

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR251850	<p>Generating serialVersionUID via quickfix hangs Eclipse</p> <p>If you try to fix the missing serialVersionUID warning via the quickfix (offered after pressing Ctrl-1), it can cause Eclipse to hang. This is a known problem in Eclipse 3.1, and will be fixed in subsequent versions of Workshop for WebLogic.</p> <p>Platform: All</p> <p>Workaround: Add a serialVersionUID to the class manually or ignore the warning.</p>
CR260226	<p>Title: After Upgrade to Beehive Tags tagid Attribute Is Not Unique Across Pageflow Portlets</p> <p>Description: The tagid attribute on netui tags isn't unique across pageflow portlets after migrating to the Beehive tags unless you add a scriptContainer tag. Adding the scriptContainer tag forces a unique tagId to be generated.</p> <p>Platform: All</p> <p>Workaround: Use a scriptContainer tag as shown below to get ID attributes rewritten with unique values.</p> <pre><% @ taglib uri="http://beehive.apache.org/netui/tags-html-1.0" prefix="netui" %> <netui:scriptContainer> <netui:anchor action="myAction" tagId="myId">some text</netui:anchor> </netui:scriptContainer></pre>
CR264808	<p>Error is displayed in the New Server wizard even if an 8.1 domain is successfully upgraded</p> <p>If a user selects a version 8.1 domain in the New Server wizard, a hyperlink is displayed to allow user to launch the Domain Upgrader. Upon successfully upgrading the domain, the state of wizard is not refreshed under some circumstances. The old error message on the top of the wizard and the upgrade hyperlink remain, and the Next button is not enabled.</p> <p>Platform: All</p> <p>Workaround: The problem can be worked around by re-selecting the text in the domain home combo box.</p>
CR265853	<p>Eclipse hangs if you put a breakpoint in an interface</p> <p>The Eclipse debugger allows the user to place a breakpoint on an interface declaration. Sometimes, when this breakpoint fires, Eclipse will hang. This is a known problem in Eclipse 3.1, which Workshop for WebLogic version 9.2.1 is based upon.</p> <p>Platform: All</p> <p>Workaround: Place breakpoints only on, or in, actual method implementations, never on a method declaration found in an interface file.</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR267837	<p data-bbox="243 383 1174 447">Javadoc attachments for Workshop for WebLogic libraries can be broken if Workshop for WebLogic is not installed into the default directory under <BEA-HOME></p> <p data-bbox="243 453 1174 609">During installation there is an option to specify the directory into which Workshop for WebLogic version 9.2.1 is installed. If you specify something other than the default (which is “workshop92”), then the Javadoc for Workshop for WebLogic code such as the service control or timer control cannot be found by the IDE. Thus, actions like Shift-F2 to show the documentation for the classes will not work</p> <p data-bbox="243 614 1174 649">Platform: All</p> <p data-bbox="243 654 1174 881">Workaround: Add Javadoc attachments to the jars contained in those libraries manually. For example, to add a javadoc attachment to the base Workshop for WebLogic controls jar, go to Windows > Preferences > WebLogic > J2EE Libraries. Select weblogic-controls-1.0 and click Edit. Expand each jar in the Classpath Contribution: tree, right-click the Javadoc location node, and select Edit. Next set the Javadoc location path: to <BEA-HOME>/<your-workshop-dir>/workshop4WP/docs/api, where <your-workshop-dir> is the non-default workshop directory name you entered during install.</p>
CR267912	<p data-bbox="243 887 1174 921">Source Not Found when debugging JDK classes</p> <p data-bbox="243 927 1174 996">In some cases while debugging an application the source for the JDK classes cannot be found, and will result in a “Source Not Found” page for the class.</p> <p data-bbox="243 1001 1174 1036">Platform: All</p> <p data-bbox="243 1041 1174 1111">Workaround: The workaround is to add a Source attachment manually. This can be done either at the workspace preference level, or if already running a debug session it can be done right there.</p> <p data-bbox="243 1116 1174 1255">If not yet in a debug session, go to the Windows > Preferences > Java > Installed JREs preference page. Select the jdk150_04 JRE and click Edit. Deselect “Use default system libraries” on the Edit JRE dialog. Open the node for rt.jar and select the Source attachment node. Click Edit, then select External File... and navigate to <BEA-HOME>/jrocket90_150_04/ and select src.zip.</p> <p data-bbox="243 1260 1174 1366">If you are already in a debug session and you get an editor page indicating “Source Not Found” for a JDK class, select the “Edit Source Lookup Path...” button. Click Add, then select “External Archive”. In the file dialog navigate to <BEA-HOME>/jrocket90_150_04/ and select src.zip.</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR269490	<p>Threads display “(may be out of sync)” in the debugger</p> <p>When debugging an application on the server, occasionally when saving code changes, you may see the following displayed in the debugger window on one or more of the threads: “(may be out of sync)”. If this occurs, you will need to republish your application in order to resync the server with the code you have just entered. This is a known problem with Eclipse 3.1.</p> <p>Platform: All</p> <p>Workaround: Republish the application to the server.</p>
CR271247	<p>Unexpected server error may require IDE restart</p> <p>Some uncontrolled server errors or terminations may cause publish status and publish operations errors. For example, if an Out Of Memory condition occurs in the server process, both Workshop for WebLogic and the server may require a restart.</p> <p>Platform: All</p> <p>Workaround: Restart Workshop for WebLogic and restart the server.</p>
CR272082 CR273414 CR272245	<p>JSP tag variables cannot be resolved in the JSP editor</p> <p>Due to a problem in Eclipse, some JSP tags (such as <auth:login> and <portlet:actionURL>) and variables declared from JSP tags are marked as containing an error when they are actually correct; although no error actually exists, Eclipse will not publish (deploy) the application.</p> <p>Platform: All</p> <p>Workaround: If this situation occurs, you must turn off JSP validation before publishing. Leave JSP validation on until you have fixed any problems except those caused by these tags; before deploying, select Window > Preferences, select Validation in the tree, and uncheck the JSP Syntax Validator check box.</p>
CR277475	<p>The console used for server startup on Linux may not display multi-byte characters correctly</p> <p>Because the IDE uses xterm to start the server, multi-byte characters may not display correctly in the console.</p> <p>Platform: Non-English Linux Operating Systems</p> <p>Workaround: Start the server from the command line in a console that is correctly configured for the language in use, outside of the IDE. After creating the server definition in the IDE, it will be able to connect to the already running server.</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR280928	<p>Tuxedo Control is no longer available with Workshop for WebLogic version 9.2.1</p> <p>Tuxedo Control will no longer be available with Workshop for WebLogic.</p> <p>Platform: All</p> <p>Workaround: Refer to the Tuxedo Control Migration white paper provided in PDF format. The PDF provides migration options from the Tuxedo Control to other alternatives.</p>
CR282777	<p>The first time you run Project > Clean after importing a project, Workshop for WebLogic may not clean all of the files from the .apt_src directory</p> <p>When a user imports a project including build directories such as .apt_src, all files in that directory might not get removed the first time a clean is performed. This only happens the first time, subsequent cleans correctly handle the generated directories.</p> <p>Platform: All</p> <p>Workaround: Manually delete the files in the .apt_src directory when performing a clean the first time after import. Once deleted, the new files that go into that directory are correctly handled by the clean action.</p>
CR283022	<p>Xbean wrapper classes generated for a ServiceControl must not be visible to the target JWS.</p> <p>When generating Xbean types for a Service Control, there are a limited number of situations where the wrapper type from the WSDL is exposed as a type in the Service Control (e.g. when there are multiple occurrences of the same Document type in the same operation signature). If the generated types jar is visible to the target JWS classloader, a duplicate type error will be thrown during deployment.</p> <p>Platform: All</p> <p>Workaround: Place Service Controls that use wrapper types in a separate project than the Service that they call. If the ServiceControl classes are loaded from a utility project, then the Service Control and the target Service must be in separate applications.</p>
CR283457	<p>XBeans are not supported as parameters or return type for doc/lit/bare operations.</p> <p>Use of Xbeans as a parameter or return type with doc/lit/bare bindings is not supported in operations or callbacks and will result in a failure during deployment.</p> <p>Platform: All</p> <p>Workaround: Use doc/lit/wrapped for services that use XBeans as parameters or return types.</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR283533	<p>Not all Beehive APIs shipping with Workshop for WebLogic are final</p> <p>Workshop for WebLogic ships an SVN snapshot of Apache Beehive (post v1.0.1). There are certain APIs that were introduced into Beehive at Apache, post v1.0.1, but have not officially shipped with a release from Apache.</p> <p>The following APIs should not be considered frozen.</p> <ul style="list-style-type: none"> • Session mutex, HttpSessionMutexListener... There is a new design change to the way NetUI does user-wide locking. An optional HttpSessionListener has been implemented that will create a session-wide mutex object that can be used to serialize a change in the session. • Plug points into low level features to handle AJAX requests (.xhr) for the Tree and DivPanel in NetUI... In the 1.0 release of Beehive, the NetUI Tree and DivPanel are both AJAX enabled. There are some new plug points to allow more advanced handling and routing of the requests. The API changes include: <ul style="list-style-type: none"> – The URLRewriter getAjaxUrl() method can be used in URL rewriting to provide information for an AJAX request. – Chain of Responsibility (CoR) / command pattern for servicing AJAX requests. This is a new command handling infrastructure that can be used to handle requests to render markup or data to a client. At a high level, this is done by implementing command handler classes that are composed together using the Chain of Responsibility (CoR) pattern and including the commands in the NetUI configuration. <p>Platform: All</p> <p>Workaround: None</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR283697	<p>Excessive slowness of XMLBeans Builder for Large XSD/WSDL Files</p> <p>Processing large (i.e. ~1MB) schema files in Workshop for WebLogic with the XMLBeans Builder can result in unacceptably long build times due to the performance of both the XMLBeans compiler and the Eclipse Java builder.</p> <p>Platforms: All</p> <p>Workaround: XMLBeans compiler performance: XMLBeans compiler performance can be improved by disabling assertions for the XMLBeans code. Assertions can be disabled by adding the line "-da:org.apache.xmlbeans..." to the file workshop92/workshop4WP/workshop4WP.ini. This change will have the most dramatic effect if the XMLBeans Builder is not used to generate Java and XMLBeans files (e.g. in simple ALSB XQuery Mapper projects).</p> <p>Java builder performance: The incremental nature of the XMLBeans and Java Builders ensures that subsequent build cycles only process the modified elements of the schema (and generated Java) and should thus complete in a fraction of the original time, however, a clean rebuild of the project will incur the full processing cost. Therefore, if XMLBeans source is required and performance of the incremental XMLBeans builder proves problematic, you may want generate the XMLBeans Jar file once outside of Workshop for WebLogic via the commandline build and then reference the generated Jar in your application.</p>
CR286141	<p>WebLogic EJB project properties not visible from generated Ant build scripts</p> <p>Description: The "Jar settings" properties and EJBC flags that can be set on WebLogic EJB projects via Project->Properties->WebLogic EJB are only used when the IDE build executes; these settings are not visible when exported Workshop Ant build scripts are executed.</p> <p>Platforms: All</p> <p>Workaround: Prior to building the project with the exported Ant script, specify all desired "Jar settings" properties using weblogic.ejbgen.JarSettings annotations in your EJB Java source files and add and desired EJBC flags directly to the build script where "weblogic.ejbcb" is executed.</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR294727	<p>Users may need to restart IDE if internet connection is lost during installation of 9.2.1 upgrade</p> <p>During the installation phase of the 9.2.1 upgrade, if the internet connection is lost, users may see an error that a JAR file was not completely downloaded:</p> <p style="text-align: center;">Unable to complete action for feature "BEA Workshop for WebLogic Platform" due to errors. Unable to retrieve remote reference ... [Unexpected end of file from server] Unable to retrieve remote reference</p> <p>Platforms: All</p> <p>Workaround: Users should verify their internet connection and restart the IDE prior to repeating the upgrade process.</p>
CR295097	<p>Cannot generate XML types from order.xsd</p> <p>Generating XML types for a .xsd file returns an error, if the .xsd file refers to another .xsd file through normal relative path (file protocol).</p> <p>Platforms: All</p> <p>Workaround: None</p> <p>Note: This problem has been resolved in Workshop for WebLogic Platform 10.0.</p>

Table 1 Known Limitations in BEA Workshop for WebLogic Platform Version 9.2.3

Problem ID	Description
CR301661	<p>Buffered methods on ServiceControl may fail over JMS protocol</p> <p>Buffered operations on a ServiceControl must be void. However the Message Exchange Pattern (MEP) in the underlying WSDL can be either request/response (with an empty response), or oneway (a request with no response).</p> <p>In the case of a request/response MEP over JMS, the presence of @MessageBuffer will cause the request to deadlock and eventually timeout. The following warning message will generally be produced:</p> <pre>Potential blocking operation {http://someNamespace}someOperation: a synchronous request/response invocation within a transaction using the JMS transport can cause deadlocks. Please refer to WebLogic documentation for details.</pre> <p>The resulting error message will include text similar to:</p> <pre>javax.xml.rpc.soap.SOAPFaultException: Failed to receive message java.io.IOException: Request timed out</pre> <p>Note: this only occurs when the transport protocol for the request is JMS.</p> <p>Platforms: All</p> <p>Workaround: If you can influence the design of the target JWS, having the JWS operation annotated with @Oneway will direct that the underlying MEP be oneway, and will avoid this situation. If you can not influence the design of the target JWS, then the workaround is to add the TransactionAttribute annotation to the ServiceControl operation:</p> <pre>@MessageBuffer @TransactionAttribute(TransactionAttributeType.NOT_SUPPORTED) public void voidMethod();</pre> <p>Note that the presence of the @TransactionAttribute will not change the transactional behavior of actions that occur within the calling application.</p>
CR311240	<p>Run time server points to the previous version</p> <p>After upgrading a Workshop for WebLogic Platform application from 9.2 to 9.2 MP1, the run time server points to the previous version instead of the latest version.</p> <p>Platforms: Windows</p> <p>Workaround: In Workshop, select Window > Preferences > Server > Installed Runtime. Select the appropriate runtime.</p>

Resolved Problems for Workshop for WebLogic Version 9.2.3

Table 2 Resolved Problems for Workshop for Weblogic Version 9.2.3

Problem ID	Description
CR286848	<p>Transactions were not propagated within nested controls. This problem was observed in transactional web services with nested controls such as a web service calling a custom control that, in turn, called a JDBC control.</p> <p>This problem has been resolved.</p>
CR307168	<p>The generation of XMLBean using the JAX-RPC option on XSD and WSDL files located in the WSDL folder did not work.</p> <p>This problem has been resolved.</p>
CR325551	<p>Regenerating XMLBeans during iterative development used to return <code>NoClassDefFoundError</code>.</p> <p>This problem has been resolved.</p>
CR328084	<p>When accessing a service control that was deployed as a shared library, <code>java.net.MalformedURLException: unknown protocol: jarnocache</code> was returned.</p> <p>This problem has been resolved.</p>
CR329594	<p>Publishing of a web service or a control that accessed a web service used to fail when Workshop for WebLogic Platform 9.2.0 was upgraded to 9.2.1 or later using Platform Upgrade Installer.</p> <p>This problem has been resolved.</p>
CR340998	<p>The <code>PageFlowViewHandler.renderView</code> method was handling only <code>HttpServletRequest</code>. For other types of request objects such as <code>RenderRequest</code> for JSR 168 JSF bridge in portal, the system returned <code>ClassCastException</code>.</p> <p>This problem has been resolved. The <code>PageFlowViewHandler.renderView</code> method can now handle any type of request.</p>
CR321676	<p>When attempting to create a web services project using generic types in the web services, Workshop returned a <code>java.lang.AssertionError</code> error and stopped responding.</p> <p>This problem has been resolved.</p>

Table 2 Resolved Problems for Workshop for Weblogic Version 9.2.3

Problem ID	Description
CR325594	Republishing of projects from IDE was delayed due to invocation of an unnecessary method. This problem has been resolved.
CR334539	Publishing of a large application used to take a long time. This problem has been resolved. The processing time has been improved by caching JAR files on the classloader.
CR303875	When the <code>java.util.List</code> type was used for Workshop parameters, the IDE did not generate the correct WSDL or Types JAR File for JAX-RPC Types from JWS. This problem has been resolved.

Resolved Problems for Workshop for WebLogic Version 9.2.2

Table 3 Resolved Problems for Workshop for Weblogic Version 9.2.2

Problem ID	Description
CR303749	A WebLogic Workshop application upgrade from 8.1.x to 9.2.x used to raise an <code>IllegalStateException</code> . This problem has been resolved.
CR307334	The WebLogic Server patches directory <code>weblogic_patches/classes</code> under <code>\$BEA_HOME/weblogic92/workshop</code> was not created by the upgrader as expected. This problem has been resolved.
CR308223	Publishing of a large application used to take a long time. This problem has been resolved. The processing time is improved by caching JAR files on the classloader.
CR308760	A web service could not be deployed when relative path with the “..” segment was used to import a schema. This problem has been resolved.

Table 3 Resolved Problems for Workshop for Weblogic Version 9.2.2

Problem ID	Description
CR309401	A <code>ModuleException</code> used to occur when deploying a web service using Apache XML Beans as the binding type. This exception used to occur because XML Bean objects were not successfully identified as XML Bean artifacts. This problem has been resolved.
CR309521	The <code><netui:errors/></code> tag was not compatible with 9.2 page flow framework and used to raise a <code>JspException</code> . This problem has been resolved.
CR309935	Processing of <code>EventSets</code> with inner classes used to fail in Eclipse. This problem has been resolved.
CR310000	The <code>ServiceControlException.hasSoapFault()</code> used to return <code>false</code> instead of <code>true</code> , if <code>SOAPFault</code> was wrapped in a <code>RemoteException</code> . This problem has been resolved.
CR311377	<code>WeblogicPageFlowEventReporter</code> used to raise a <code>NullPointerException</code> when requests were processed for non-existent actions in the Struts configuration file. This problem has been resolved.
CR313766	An <code>InvokeException</code> used to occur when calling <code>SecureNewsService</code> of the <code>SamplesApp</code> because web service client artifacts were not generated for the second level package, if web services with callbacks were found at the first level package. This problem has been resolved.
CR317631	WLW 9.2 IDE used to crash when clicking the Browse Server EJBs button to obtain a list of integration EJBs. This problem has been resolved.

Resolved Problems for Workshop for WebLogic Version 9.2.1

Table 4 Resolved Problems for Workshop for Weblogic Version 9.2.1

Problem ID	Description
CR279626	<p>While attempting to compile a service control fronting a JWS with @usewlv81types using a WebLogic 9.2 WSDL, the compilation used to fail.</p> <p>This problem has been resolved. A message is displayed to regenerate the Service Control if WebLogic 9.2 WSDL is used.</p>
CR287018	<p>A deadlock used to occur in the NetUI framework with a direct request to a JSP that includes another JSP.</p> <p>This problem has been resolved.</p>
CR288871	<p>While building and redeploying projects, some files (specifically .jar files) are cached by the classloader. This causes problems when JARs are deleted or expected to be deleted, because they are cached (and not cleaned up, as required).</p> <p>This problem has been resolved. The clean and redeploy functionality works as expected.</p>
CR290847	<p>The EJB Control Assembler was adding EJB references only to session EJBs. Therefore, it was not possible to use EJB controls with an @ejbLink annotation from within an Entity EJB.</p> <p>This problem has been resolved. The EJB Control Assembler now adds EJB references to entity EJBs as well as session EJBs for EJB Controls with an @ejbLink annotation.</p>
CR290852	<p>An upgraded WebLogic Platform 8.1 Service Control was not being allowed to start a conversation on an upgraded WebLogic Platform 8.1 web service from a (server-side) buffered method.</p> <p>This problem has been resolved. An upgraded WebLogic Platform 8.1 Service Control is allowed to start a conversation on an upgraded WebLogic Platform 8.1 web service from a (server-side) buffered method.</p>
CR293965	<p>Problem with the coalesce function caused incorrect timeout behavior.</p> <p>This problem has been resolved. Coalesce functions now work as expected.</p>
CR293966	<p>The SoapFault.getDetailContents() method used to return null.</p> <p>This problem has been resolved. The SoapFault.getDetailContents() method returns an array of size 0 when there are no details, instead of returning null.</p>

Table 4 Resolved Problems for Workshop for Weblogic Version 9.2.1

Problem ID	Description
CR294591	<p>The EJB Timer Control JNDI name was not being prefixed with the application name. Because of that, versioned deployment of applications using the EJB Timer Control was not supported.</p> <p>This problem has been resolved. The EJB Timer Control JNDI name is prefixed with the application name to support versioned deployment.</p>
CR295111	<p>The MessageBuffer Message-Driven Bean (MDB) deployment used to fail because there was no submoduletargets specified for the application-scoped JMS module containing the MDB.</p> <p>This problem has been resolved. A JMS server sub-deployment function is added to the weblogic-controls library module in the BEA Workshop for WebLogic Platform domain template to enable deployment of MDBs.</p>
CR295898	<p>The <code>ServiceControlException.getMessage()</code> method was not returning the fault XML, as expected.</p> <p>This problem has been resolved. The fault XML is returned, as expected.</p>
CR297583	<p>The TimerControl timer was not getting activated immediately when the timeout was set to 0 or was not specified.</p> <p>This problem has been resolved. The TimerControl timer gets activated, as expected, if the timeout is set to 0, or is not specified.</p>
CR299479	<p>The default timeouts for WebLogic Server deployment (5 minutes), startup and shutdown (8 minutes) were insufficient for some very large applications.</p> <p>This problem has been resolved. These timeouts can now be set in milliseconds using the following global system properties: <code>weblogic.deployment.timeout</code>, <code>weblogic.startserver.timeout</code>, <code>weblogic.stopsserver.timeout</code>.</p> <p>If these properties are not specified, the default timeout (5 minutes for deployment and 8 minutes for startup and shutdown) are applicable.</p>

