Oracle® Fusion Middleware

Oracle WebLogic Server Release Notes

11*g* Release 1 (10.3.6)

E58096-03

September 2016

This document describes all known issues for this release of Oracle WebLogic Server.

Oracle Fusion Middleware Oracle WebLogic Server Release Notes, 11*g* Release 1 (10.3.6)

E58096-03

Contents

Preface

This document describes all know issues with Oracle WebLogic Server 11*g* Release 1 (10.3.6).

Audience

This document is intended for all users of Oracle WebLogic Server 11*g* Release 1 (10.3.6).

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following Oracle resources:

- Oracle WebLogic Server Online Documentation library
- Oracle Technology Network at `http://www.oracle.com/technetwork/index.html`.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

This chapter introduces Oracle Fusion Middleware Release Notes for Oracle WebLogic Server 11*g* Release 1 (10.3.6). It includes the following topics:

- Section 1.1, "Latest Release Information"
- Section 1.2, "Purpose of This Document"
- Section 1.3, "System Requirements and Specifications"
- Section 1.4, "Certification Information"
- Section 1.5, "Licensing Information"

1.1 Latest Release Information

This document is accurate at the time of publication. Oracle will update the release notes periodically after the software release. You can access the latest information and additions to these release notes on the Oracle Technology Network at:

`http://www.oracle.com/technetwork/indexes/documentation/index.html`

1.2 Purpose of This Document

This document contains the release information for Oracle WebLogic Server 11*g* Release 1 (10.3.6). It describes differences between the software and its documented functionality.

Oracle recommends that you review its contents before installing or working with the product.

1.3 System Requirements and Specifications

Oracle WebLogic Server installation and configuration will not complete successfully unless users meet the hardware and software pre-requisite requirements before installation.

For more information, see the *Installation Guide for Oracle WebLogic Server*.

1.4 Certification Information

For certification information, refer to the certification matrix at `http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html`.

1.5 Licensing Information

Licensing information for Oracle WebLogic Server is available at:

http://shop.oracle.com/

2

Oracle WebLogic Server Issues

This chapter describes issues associated with Oracle WebLogic Server. It includes the following topics:

- Section 2.27, "RMI-IIOP Issues and Workarounds"
- Section 2.28, "Security Issues and Workarounds"
- Section 2.29, "SNMP Issues and Workarounds"
- Section 2.30, "Spring Framework on WebLogic Server Issues and Workarounds"
- Section 2.31, "System Component Architecture (SCA) Issues and Workarounds"
- Section 2.32, "Upgrade Issues and Workarounds"
- Section 2.33, "Web Applications Issues and Workarounds"
- Section 2.34, "WebLogic Server Scripting Tool (WLST) Issues and Workarounds"
- Section 2.35, "Web Server Plug-Ins Issues and Workarounds"
- Section 2.36, "Web Services and XML Issues and Workarounds"
- Section 2.37, "WebLogic Tuxedo Connector Issues and Workarounds"
- Section 2.38, "Documentation Errata"

Note: For a list of bugs that are fixed in WebLogic Server 11*g* Release 1 (10.3.6), enter the following document ID in the Search Knowledge Base field. You must enter the entire document ID.

1302753.1

2.1 General Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.1.1, "Multi-Byte Characters Display Incorrectly in Filenames When Using Safari"
- Section 2.1.2, "Oracle WebLogic Server Version Number"
- Section 2.1.3, "Oracle ojdbc14.jar File Has Been Changed to ojdbc6.jar"
- Section 2.1.4, "Strong Password Enforcement May Cause Issues With WLST Offline Scripts"
- Section 2.1.5, "In Turkish Locale, MDS Initialization Fails"
- Section 2.1.6, "Administration Server Reports a 'Too Many Open Files' Message on the EM Console"
- Section 2.1.7, "Availability of Sun JDK 6 U35-B52 for 10.3.5.0 Oracle WLS Generic Installation"
- Section 2.1.8, "Availability of Fixes for IBM JDK 6.0 SR16 FP3 or JDK 7.0 SR8 FP10"

2.1.1 Multi-Byte Characters Display Incorrectly in Filenames When Using Safari

Platform: All

When using the Safari browser to download content, if a filename contains multi-byte characters, the characters are displayed as '------' in the filename.

Workaround

Set `UseHeaderEncoding` to `true` on the Managed Server. Use the following WLST commands to do so:

```
connect("admin_name", "admin_password", "t3://localhost:port")
edit()
startEdit()
cd("Servers/server_name/WebServer/server_name")
set("UseHeaderEncoding", "true")
save()
activate()
exit()
```

2.1.2 Oracle WebLogic Server Version Number

Platform: All

Oracle Fusion Middleware 11*g* contains Oracle WebLogic Server 11*g*. The version number of Oracle WebLogic Server is 10.3.6.

2.1.3 Oracle ojdbc14.jar File Has Been Changed to ojdbc6.jar

Platform: All

The Oracle `ojdbc14.jar` file has been changed to `ojdbc6.jar`, for use with JDK 5 or 6. As a result, any explicit references you make to `ojdbc14.jar` must be changed to `ojdbc6.jar`.

2.1.4 Strong Password Enforcement May Cause Issues With WLST Offline Scripts

Platform: All

With the implementation of strong password enforcement (8 character minimum with one numeric or special character) in this release of WebLogic Server, existing scripts could potentially encounter issues.

Workaround

Use either of the following workarounds to bypass the new password restrictions.

- Set the `BACKWARD_COMPAT_PW_CHECK` environment variable to `true`.
- Include the `-Dbackward.compat.pw.check=true` option when invoking WLST.

Oracle recommends that you change passwords to comply with the new password requirements, as this variable and option will be removed in a future release of WebLogic Server.

2.1.5 In Turkish Locale, MDS Initialization Fails

Platform: All

Any applications that use an MDS repository cannot be deployed or run with the JAXB version bundled with WebLogic Server as null values are returned for attributes named `id`.

Workaround

Start the server in English locale.

2.1.6 Administration Server Reports a 'Too Many Open Files' Message on the EM Console

Platform: Linux

The WebLogic Server Administration Server reports a `Too Many Open Files` message on the Enterprise Manager (EM) console when the maximum number of file descriptors configured for the Administration Server is less than 65535.

Workaround

Execute the following command to determine the maximum number of file descriptors currently configured:

```
cat /proc/sys/fs/file-max
```

If the value is less than 65535, perform the following steps:

1. Edit the file /etc/security/limits.conf with root permission:

   ```
   > sudo vi /etc/security/limits.conf
   ```

2. Append the following two lines, using a value of 65535 or greater:

   ```
   *                soft    nofile          65535
   *                hard    nofile          65535
   ```

3. Start a new terminal session.
4. Execute the `limit descriptors` command to verify that descriptors has been increased to the specified value (at least 65535).

   ```
   > limit descriptors
   descriptors  65535
   ```

2.1.7 Availability of Sun JDK 6 U35-B52 for 10.3.5.0 Oracle WLS Generic Installation

Platform: IBM AIX

Sun JDK 1.6.0.U35-B52 version is required for Oracle WebLogic Server 10.3.5.0 (PS4) generic installation on Linux x86-64, Microsoft Windows x64 (64-Bit), and Oracle Solaris platforms.

The mentioned version of JDK is not available for download from the Oracle Web site:

http://www.oracle.com/technetwork/indexes/downloads/index.html

Complete the following steps to download the required JDK version:

1. Go to My Oracle Support:

   https://support.oracle.com

2. Click the **Patches & Updates** tab.
3. Enter patch 12346791 in the **Patch Name or Number** field, under Patch Search.
4. Click **Search**.
5. Select and download the patch for the required platform by following the instructions in the `README` file included with the patch.

2.1.8 Availability of Fixes for IBM JDK 6.0 SR16 FP3 or JDK 7.0 SR8 FP10

Platform: IBM AIX and zLinux

If WebLogic Server is configured with IBM JDK 6.0 Service Refresh 16 with Fix Pack 3, or JDK 7.0 Service Refresh 8 with Fix Pack 10, WebLogic Server may not function correctly if you do not install the fix for APAR IV71293 on that JDK. APAR IV71293 is described at the following URL:

http://www-01.ibm.com/support/docview.wss?uid=swg1IV71293

You can obtain the fix for APAR IV71293 using either of the following methods:

- Go to the IBM Support Portal, where you can open a Problem Management Record (PMR) using the Service Request (SR) tool and request the fix for APAR IV71293.
- Go to the IBM Java for ISVs portal and download the fix as follows:
  1. Enter the following URL in a browser:

     https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-ibmjavaisv
  2. Log in with your universal IBM ID. (You may need to contact IBM support for your entitlement agreement key.)
  3. Download one of the following fixes, as appropriate for your IBM JDK release:
     - For JDK 6.0 SR16 FP3: pap6460sr16fp3ifix-20150323_01 (SR16FP3 + IV71293)
     - For JDK 7.0 SR8 FP10: pap6470sr8fp10ifix-20150323_01 (SR8FP10 + IV71293)

2.2 Administration Console Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.2.1, "Cached JDBC Information is not Displayed"
- Section 2.2.2, "Pressing Browser Back Button Discards Context"
- Section 2.2.3, "Unsupported Work Manager Configurations Can Be Created"
- Section 2.2.4, "Server Status Table Reflects Inconsistent Information"
- Section 2.2.5, "Exceptions When Defining a Security Policy for an EJB"
- Section 2.2.6, "Administration Console Does Not Always Reflect External Changes Made in a Deployment Plan"
- Section 2.2.7, "Oracle OCI Driver Support"
- Section 2.2.8, "Data Takes a Long Time to Display on the Metric Browser Tab"

2.2.1 Cached JDBC Information is not Displayed

Platform: All

Information about cached JDBC statements is not displayed on the JDBC Monitoring pages.

2.2.2 Pressing Browser Back Button Discards Context

Platform: All

After a page flow completes in the Administration Console, it forwards to a different page, typically a table.

Pressing the browser Back button at this point results in an attempt to load the last JSP file in the completed assistant. At this point, all of the context for this assistant is discarded.

Workaround

Oracle recommends that you do not use the browser Back button to step back into an assistant once changes are cancelled or finished, and that you do not go back to a previous step in an assistant. Instead, use the navigation links and buttons in the Administration Console.

2.2.3 Unsupported Work Manager Configurations Can Be Created

Platform: All

The Administration Console permits the creation of Work Manager configurations that are not supported and do not function as intended. Incorrect Work Manager configurations may result in a number of exceptions being recorded in the server logs, most commonly `'Validation problems were found'` exceptions while parsing deployment descriptors.

Workaround

Follow the guidelines described in the online help for Work Manager configurations. Specifically, you can only assign one request class to any given Work Manager, and that request class must be of the same or a broader scope than the Work Manager. You should not assign an application-scoped request class to a global Work Manager, and you should not create more than one application-scoped request class for an application-scoped Work Manager.

Correcting the Work Manager configurations to match the documented constraints resolves these issues.

2.2.4 Server Status Table Reflects Inconsistent Information

Platform: All

The Server Status table on the **Cluster: Monitoring: Summary** page includes two default columns: **Primary** and **Secondary Distribution Names**. These fields do not always reflect all of the replication statistics that are collected and displayed on the **Cluster: Monitoring: Failover** page, depending on the replication scenario.

Please refer to the **Cluster: Monitoring: Failover** page for definitive information.

2.2.5 Exceptions When Defining a Security Policy for an EJB

Platform: All

When defining security policies in the Administration Console for an EJB deployment that references types defined in a separate library deployment, exceptions can be observed if that library deployment is not available to the Console.

Workaround

All library deployments should be targeted at the WebLogic Server Administration Server as well as any Managed Servers needed to support referencing applications. This will ensure that when defining policies, the Console will have access to those library deployments so that referenced types can be class-loaded as needed.

2.2.6 Administration Console Does Not Always Reflect External Changes Made in a Deployment Plan

Platform: All

The Administration Console does not always reflect external changes made in a deployment plan. If a change is made in a deployment plan outside of the Console (for example, using Workshop, editing the plan text files directly, or updating a deployment with a new plan using WLST or webLogic.Deployer) while a Console user is also viewing that deployment plan, the Console user will not see those changes.

Workaround

Navigate to a configuration page for a different deployment, then navigate back to the original deployment again.

2.2.7 Oracle OCI Driver Support

Platform: All

The Oracle OCI driver is no longer explicitly listed as a preconfigured driver type in the Administration Console.

Workaround

The Oracle OCI driver remains a supported driver for application data connectivity, consistent with prior releases of Oracle WebLogic Server. However, users must now specify all required configuration properties manually, including the data base username.

2.2.8 Data Takes a Long Time to Display on the Metric Browser Tab

Platform: All

When using Internet Explorer 7 (IE 7) to display data on the Metric Browser tab of the Monitoring Dashboard, it takes an unusually long time for the data to display, and during this time, the page is unresponsive. The amount of time it takes to display data on this tab depends on the size of the domain.

Workaround

If you need to display data on the Monitoring Dashboard > Metric Browser tab, open the Administration Console in a supported web browser other than IE 7, such as Internet Explorer 8 or greater, Firefox 3 or greater, or Safari 4 or greater.

2.3 Apache Beehive Support Issues and Workarounds

There are no known Apache Beehive Support issues in this release of WebLogic Server.

2.4 Clustering Issues and Workarounds

This section describes the following clustering issue and workaround:

- Section 2.4.1, "Threads Are Blocked on Cluster Messaging in Unicast Mode"

2.4.1 Threads Are Blocked on Cluster Messaging in Unicast Mode

Platform: linux

When using Unicast mode for cluster communication, many threads are blocked on cluster messaging, which may result in cluster members having difficulty sending heartbeat messages. In this situation, some cluster members drop out from the cluster and may take some time to rejoin the cluster.

Workaround

Set the following system property to resolve this issue:

```
-Dweblogic.unicast.HttpPing=true
```

2.5 Configuration Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.5.1, "ASProvWorkflowException Occurs When Creating a WebLogic Domain"
- Section 2.5.2, "Use the -Dfile.encoding Property When Running WLST in a Non-English Locale"
- Section 2.5.3, "Configuration Tools Can Fail If WebLogic Installation Path Contains Spaces"
- Section 2.5.4, "Directory For a Non-Existent Server Name Is Created"
- Section 2.5.5, "Abnormal Behavior in Terminal Window After Entering WebLogic Password"
- Section 2.5.6, "Creating and Updating Domains Takes Too Long"
- Section 2.5.7, "Password Field Is Not Editable When Configuring a New Domain"

2.5.1 ASProvWorkflowException Occurs When Creating a WebLogic Domain

Platform: All

In rare cases, if your installation environment contains existing JAVA_OPTIONS prior to starting a Fusion Middlware product installation, these may cause an `ASProvWorkflowException`, preventing the domain from being created.

Workaround

Prior to starting the Fusion Middleware product installation, clear the existing JAVA_OPTIONS. If you have an applicagtion in the environment that use these JAVA_OPTIONS, the applications may not work after clearing the options. In this case, save the existing JAVA_OPTIONS to a text file and investigate alternatives for running your other application.

2.5.2 Use the -Dfile.encoding Property When Running WLST in a Non-English Locale

Platform: MS Windows

WLST can be run with localized messages by setting the desired locale. You should be aware of the following issue when running WLST in a non-English locale.

On Windows operating systems, if a DOS command window's active code page is different from the system's local (ANSI) code page, you must add the `-Dfile.encoding=<`*`DOS window's active code page`*`>` property to the WLST process when starting WLST via a DOS command window. This changes the default character set for the Java process. For example:

- The active code page for a DOS window is 850. This can be achieved by issuing the `chcp` command in the WLST command window.
- The system's local (ANSI) code page is 1250. You can determine the system's local code page by viewing the value of the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\NLS\CodePage\ACP key in

the Windows registry. Files that are created by standard Windows editing tools (such as Notepad or Wordpad) are encoded in this way.

In this situation, you can start WLST as follows:

```
set WLST_PROPERTIES="-Dfile.encoding=cp850"
```

```
$WL_HOME%\wlserver_10.3\common\bin\wlst.cmd
```

2.5.3 Configuration Tools Can Fail If WebLogic Installation Path Contains Spaces

Platform: Microsoft Windows

On some Microsoft Windows platforms, the WebLogic configuration tool commands (including `wlst`, `config`, `pack`, and `unpack`) can fail if the WebLogic installation path contains a space. In this case, the command may fail with a `java.lang.ClassNotFoundException`, where the class is derived from the portion of the installation path after the space. The commands fail when short file name generation has been disabled in the Windows registry.

Workaround

You must enable short name generation in the Windows registry in order for spaces to be properly handled by the configuration tools. To enable short name generation:

1. Run `regedit`.
2. Navigate to the HKEY_LOCAL_ MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem folder.
3. Double-click `NtfsDisable8dot3NameCreation` and set its value to `0`.
4. Reboot for the change to take effect.

2.5.4 Directory For a Non-Existent Server Name Is Created

Platform: Linux

If you attempt to connect to the WebLogic Server Administration Server with a non-existent server name, a directory for the non-existent server name is created under the *`domain_name`*`/servers` directory.

Workaround

Specify a valid server name when connecting to the Administration Server.

2.5.5 Abnormal Behavior in Terminal Window After Entering WebLogic Password

Platform: Linux

After pressing Ctrl-C to terminate the `startManagedWebLogic.sh` process immediately after entering the WebLogic password, abnormal behavior may be experienced in the terminal window. For example, when pressing Return, the prompt is tabbed instead of going to the next line, and any characters that are entered at the prompt are not displayed in the terminal.

Workaround

Either close the current xterm and start a new one, or enter `stty echo` into the xterm.

2.5.6 Creating and Updating Domains Takes Too Long

Platform: Linux

It can take a long time to create or update WebLogic Server domains when:

- Installing WebLogic Server on UNIX or Linux operating systems if the Server Examples are included in the installation.
- Using the WebLogic Server Configuration Wizard to create or update a domain.
- Using WLST to create or update a domain.

Workaround

Set the CONFIG_JVM_ARGS environment variable to the following value:

```
-Djava.security.egd=file:/dev/./urandom
```

2.5.7 Password Field Is Not Editable When Configuring a New Domain

Platform: Linux

On Linux systems, when creating a new domain in the Oracle Fusion Middleware Configuration Wizard, the **Password** and **Confirm Password** fields are sometimes not editable, and you cannot enter a password to create a domain.

Workaround

There are two ways to work around this issue:

- To work around the issue each time it happens, click the Close Window **X** button in the upper right corner of the Configuration Wizard. In the confirmation dialog that appears, click **No** to return to the Configuration Wizard. You can then enter and confirm the password for the domain.
- To fix this issue permanently:
  1. Kill all scim processes. For example:

     ```
     kill `pgrep scim`
     ```

  2. Modify (or create) the file `~/.scim/config` to include the following line (case-sensitive):

     ```
     /FrontEnd/X11/Dynamic = true
     ```

  3. If you are running VNC, restart the VNC server.
  4. Run the Configuration Wizard again.

2.6 Connector (Resource Adapter) Issues and Workarounds

There are no known Connector (Resource Adapter) issues in this release of WebLogic Server.

2.7 Console Extensions Issues and Workarounds

There are no known Extensions issues in this release of WebLogic Server.

2.8 Core Server and Core Work Manager Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.8.1, "Threads Become Stuck While Waiting to Get a Connection"
- Section 2.8.2, "Using IPv6-Formatted Addresses"
- Section 2.8.3, "Server Cannot Be Started After a Whole Server Migration"
- Section 2.8.4, "Object State is not Retained After Renaming Field"
- Section 2.8.5, "Forcing Unicast Messages To Be Processed in Order"
- Section 2.8.6, "Servers Configured to Listen on a Host Name Are Listening on a Different Host Name After Startup"
- Section 2.8.7, "Administration Server or Node Manager Cannot Track the Status of a Managed Server"
- Section 2.8.8, "Multicast Traffic Observed to be Unreliable During or After a Network Partition"

2.8.1 Threads Become Stuck While Waiting to Get a Connection

Platform: All

When a machine that is hosting one of the Managed Servers is abruptly shut down, a network cable is pulled, or its network interface card has issues, and any server attempts communication with that managed server, threads become stuck waiting to get a connection.

Workaround

This can currently be resolved by using a private flag:

```
-Dweblogic.client.SocketConnectTimeoutInSecs
```

and setting an appropriate timeout value that will release the thread attempting to make the connection and allow the request to fail quickly.

2.8.2 Using IPv6-Formatted Addresses

Platform: All

When using an IPv6-formatted address for WebLogic Server, the URL should include square brackets ('[' and ']') for the host address. Otherwise, WLST may fail to connect to the running server.

Workaround

Add square brackets to the host address. For example:

```
t3://[fe80:0:0:0:203:baff:fe2f:59e5]:9991
```

2.8.3 Server Cannot Be Started After a Whole Server Migration

Platform: All

If the WebLogic Server Administration Server is down when a Whole Server Migration occurs for a clustered server, and the server migrates to a machine on which it was never run before, the server cannot be started on the new machine.

Workaround

Use one of the following workarounds for this issue:

- Ensure that the Administration Server is up when the server migration is being performed.
- Use a shared disk/NFS for all the migratable servers in the cluster.

2.8.4 Object State is not Retained After Renaming Field

Platform: All

When FastSwap is enabled in a Java EE application, you can make certain types of changes to Java classes during development and expect to see the change without re-deploying, with all instance states of the Java object being retained.

One type of change that does NOT retain the object state is that when a field name is changed, it is treated as follows:

- the field with old name is deleted
- the field with new name is added

Thus, in this case, any state in the old field is not carried over to the renamed field.

Using the Workshop or FastSwap ant task, you may see a `FastSwap operation completed successfully` message, even when an instance field name change causes a value reset.

Workaround

You should expect an instance value to be reset when you change a field name.

2.8.5 Forcing Unicast Messages To Be Processed in Order

Platform: All

The following conditions can cause very frequent JNDI updates, and as a result, JMS subscribers may encounter a `java.naming.NameNotFoundException`:

1. Unicast messaging is being used for cluster communication.
2. The JMS topic connection is set with `setReconnectPolicy("all").`
3. JMS durable subscribers on topic are created and removed very frequently.

Workaround

To fix this issue, a new property, `MessageOrderingEnabled`, has been added to the `ClusterMBean`. This property forces unicast messages to be processed in strict order. By default, this property is not enabled. To enable the property, add the following line manually to the `<cluster>` element in `config.xml`.

```
<message-ordering-enabled>true</message-ordering-enabled>
```

2.8.6 Servers Configured to Listen on a Host Name Are Listening on a Different Host Name After Startup

Platform: All

When using a host name to specify configuring the listen address on the WebLogic Server Administration Server or a Managed Server, machines that are configured with multiple Ethernet cards may listen on a different host name after startup. For example:

- The machine has 3 Ethernet cards
- Card 1 is mapped to `hostname1-s` (DNS registered host name)

- Card 2 is mapped to `hostname1-i` (DNS registered host name)
- Card 3 is mapped to `hostname1` (actual node's host name)
- You configure the server to listen on `hostname1`
- After starting the server, it is listening on `hostname1-s` because Windows resolves the actual node's host name to the first enabled Ethernet card address

Workaround

Use one of the following three workarounds for this issue:

1. Use the IP address, instead of the host name, as the listen address of the WebLogic Server Administration Server. On Managed Servers, use the IP address as the listen address, or configure the actual physical host name to the first Ethernet card in the machine.
2. Add the following entry to the C:\Windows\system32\drivers\etc\hosts file on the machine:

   ```
   <ip_address> <hostname>
   ```

3. Change the order of the network cards in the machine so that the card with the actual node's host name is Card 1.

2.8.7 Administration Server or Node Manager Cannot Track the Status of a Managed Server

Platform: Linux

If you start a managed server by providing an incorrect WebLogic Server Administration Server URL from the command line (that is, the Administration Server cannot be reachable at the provided URL), the managed server will start in Managed Server Independence (MSI) mode.

In this case, neither the Administration Server nor Node Manager can track the status of the managed server. The Administration Console will show the status of the managed server as UNKNOWN, but the server will actually be RUNNING in MSI mode.

2.8.8 Multicast Traffic Observed to be Unreliable During or After a Network Partition

Platform: Linux

During or after a network partition that causes a server migration to take place, multicast traffic has been observed to be unreliable. For example, one node may be receiving multicast traffic, but traffic originating from this node is not received on other nodes in the network. As a result, the migrated servers are not added to the cluster because their heartbeats were not received.

Workaround

Currently, the only known workaround is to use unicast cluster messaging.

2.9 Deployment Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.9.1, "security-permission Element is not Available in weblogic-application.xml"

2.9.1 security-permission Element is not Available in weblogic-application.xml

Platform: All

The `security-permission` element is available in the `weblogic.xml` and `weblogic-ejb-jar.xml` deployment descriptors, but is not available in the `weblogic-application.xml` descriptor. Therefore, in an Enterprise application, you can only apply security policies to JAR files that are EJBs or Web applications.

2.9.2 Extraneous String Values Interpreted as File Specification

Platform: All

The weblogic.Deployer tool interprets any extraneous string values between command-line arguments as a file specification. For example, if you enter the command:

```
java weblogic.Deployer -activate -nostage true -name myname -source
c:\myapp\mymodule
```

the tool attempts to activate a file specification named true, because the `-nostage` option takes no arguments and `true` is an extraneous string value.

2.9.3 java.lang.NoClassDefFoundError is Displayed

Platform: All

While using the WebLogic Server Administration Console with applications or EJBs deployed on a Managed Server that depend on a deployed library, you may encounter a `java.lang.NoClassDefFoundError`.

Workaround

The WebLogic Server Administration Console needs access to any shared library deployments so that Java data types and annotations can be processed. Therefore, all shared library deployments should always be targeted to the WebLogic Server Administration Server in addition to any Managed Servers or clusters.

2.9.4 The restore Method Does Not Update the DConfig Bean With Plan Overrides

Platform: All

The restore method does not correctly update the DConfig Bean with the plan overrides. For example, given the following steps:

```
  DeployableObject dObject =
   WebLogicDeployableObject.createDeployableObject(new File(appName));
DeploymentConfiguration dConfig =
   WebLogicDeploymentManager.createConfiguration(dObject);
dConfig.restore(new FileInputStream(new File(plan)));
```

the plan does not correctly override the DConfig Bean.

Workaround

Specify the plan when initializing the configuration for the application. For example:

```
helper = SessionHelper.getInstance(
    SessionHelper.getDisconnectedDeploymentManager());
helper.setApplication(app);
helper.setPlan(new File(plan));
helper.initializeConfiguration();
```

2.9.5 config-root <*directory*> not found Warning Is Displayed When Applying a Plan

Platform: All

If you use the Administration Console to make configuration changes to an application, a deployment plan will be generated. If external descriptors are generated as part of the deployment plan, they are placed in the config root plan directory. This directory will be set in the deployment plan 'config-root' attribute.

If no external descriptors are required, the config root directory will not be created, and a warning is displayed when you apply the deployment plan. This results in the following warning in the server output:

```
<Warning <WWebLogicDescriptorWL> <BEA-2156000><"config-root" C:\deployments\plan
was not found>.
```

Workaround

Create the plan directory manually.

2.9.6 Deployment Task Fails When a Large Application File Is Deployed

Platform: All

When a large application file is deployed using the `upload` option, the deployment task fails with the following error:

```
java.lang.OutOfMemoryError: Java heap space
```

To resolve this issue, a new system property, `weblogic.deploy.UploadLargeFile`, has been added. If you see this issue, include this flag in the `java` command you use to launch a deployment client.

If you are using the WebLogic Server patch releases 9.2 MP2, 9.2 MP3,10.0 MP1, 10.0 M2, 10.3, 10.3.1, 10.3.2, or 10.3.3, this flag is not needed.

2.9.7 Application State Is Not Updated If the Server Starts in MSI Mode

Platform: Linux

A managed server will start in MSI mode if the WebLogic Server Administration Server is not available when the managed server starts. If you start the Administration Server later, the managed server will connect to the Administration Server. However, the state of each application deployed to the managed server is not updated to reflect the state of the applications on the managed server. Each application's state is displayed as NEW or PREPARED in the WebLogic Server Administration Console.

Workaround

There are two workarounds for this issue:

- Start the Administration Server before starting the managed server, or
- Redeploy the application after starting the Administration Server.

2.9.8 Attempting to Redeploy an Application Fails if the Application is Already Deployed Using a Different Source File Location

Platform: Linux

If you initially deployed an application using one source file location, then attempt to redeploy the application using a new location for the source file, the deployment fails with the following exception:

```
New source location <new_source_file_path> cannot be configured deployed to
configured application, <application_name>. The application source is at
original_source_file_path. Changing the source location is not allowed for a
previously attempted deployment. Try deploying without specifying the source.
```

This is due to a WebLogic Server deployment restriction. Once you specify the source file for a deployment, you cannot change it on a redeployment.

Workaround

Undeploy the application before attempting to redeploy it using a new source file location.

2.10 EJB Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.10.1, "Primary Key in Oracle Table is CHAR"
- Section 2.10.2, "No Available Annotation That Enables Creation of a Clusterable Timer"
- Section 2.10.3, "Kodo's MappingTool Cannot Generate Schemas"
- Section 2.10.4, "Extensions to the JPA Metadata Model Can Only Be Specified Via Annotations"
- Section 2.10.5, "Lookup Method Injection Not Supported by Spring"
- Section 2.10.6, "Deserializing a JDO PersistenceManagerFactory in a Managed Environment May Fail"
- Section 2.10.7, "Indexes Not Always Created During Schema Creation"
- Section 2.10.8, "OpenJPA throws an exception when @Id fields are also annotated as @Unique"
- Section 2.10.9, "Cache Hit and Miss Counts May Rise Unexpectedly"
- Section 2.10.10, "Open JPA Tries to Create a Table Even if the Table Exists"
- Section 2.10.11, "EJB Applications Fail During Serialization"

- Section 2.10.12, "Non-Transactional Message-Driven Bean Container Can Fail to Provide Reproducible Behavior For Foreign Topics"

2.10.1 Primary Key in Oracle Table is CHAR

Platform: All

The primary key in an Oracle table is a CHAR but the query field in the SQL table is a VARCHAR2.

Workaround

Change the database schema from CHAR to VARCHAR2. Using CHAR as a primary key is not recommended for the Oracle database.

2.10.2 No Available Annotation That Enables Creation of a Clusterable Timer

Platform: All

There is no annotation for EJB3 beans or `Ejbgen` that enables creation of a clusterable timer.

Workaround

Create a weblogic-ejb-jar.xml file and put the `<timer-implementation>` element and corresponding values into the file.

2.10.3 Kodo's MappingTool Cannot Generate Schemas

Platform: All

Kodo's MappingTool cannot generate schemas for classes that use BLOBs in their primary key. BLOBs can be used in a primary key, but the schema must be defined manually. Note that support for BLOB columns in primary keys is not mandated by either the JDO or JPA specifications.

2.10.4 Extensions to the JPA Metadata Model Can Only Be Specified Via Annotations

Platform: All

Extensions to the JPA metadata model can only be specified via annotations, and not via a structure similar to the orm.xml file defined by the specification.

Workaround

To specify Kodo-specific metadata for your object model, either:

- use the Kodo-specific annotations, or
- convert your XML-based metadata to the JDO metadata format, which does support XML specification of extensions.

2.10.5 Lookup Method Injection Not Supported by Spring

Platform: All

The Weblogic Spring injection extension model doesn't support lookup method injection.

2.10.6 Deserializing a JDO PersistenceManagerFactory in a Managed Environment May Fail

Platform: All

Deserializing a JDO `PersistenceManagerFactory` in a managed environment may fail. The exception states that the `javax.jdo.PersistenceManagerFactoryClass` property is missing. Note that serializing a `PersistenceManagerFactory` should not generally be necessary in a managed environment.

2.10.7 Indexes Not Always Created During Schema Creation

Platform: All

Indexes declared at the class level are not always created during schema creation.

Workaround

Create the indexes manually after running the schema generation tools.

2.10.8 OpenJPA throws an exception when @Id fields are also annotated as @Unique

Platform: All

OpenJPA throws an exception when `@Id` fields are also annotated as `@Unique` in some databases. Database primary keys are unique by definition. Some databases implement this by creating a unique index on the column.

Workaround

Do not specify both `@Id` and `@Unique` on a single field.

2.10.9 Cache Hit and Miss Counts May Rise Unexpectedly

Platform: All

The cache hit and miss counts may rise unexpectedly when manipulating entities without version data. The extra cache access occurs when the EntityManager closes and all contained entities are detached. Entities without version fields appear to the system to be missing their version data, and the system responds by checking their version in the cache before detachment.

Workaround

Entities with version fields or other version strategies do not cause extra cache access.

2.10.10 Open JPA Tries to Create a Table Even if the Table Exists

Platform: All

When using the MySQL database, and OpenJPA is configured to automatically run the mapping tool at runtime and create tables within the default schema (for example):

```
<property name='openjpa.jdbc.SynchronizeMappings' value='buildSchema'/>
<property name='openjpa.jdbc.Schema' value='MySQL database name' />
```

OpenJPA will try to create the table even if the table already exists in the database. A PersistenceException will be thrown to indicate that the table already exists and the table creation statement fails.

Workaround

To avoid this problem, if you are using the MySQL database, don't configure OpenJPA to automatically run the mapping tool at runtime and specify the default schema at the same time.

2.10.11 EJB Applications Fail During Serialization

Platform: All

EJB applications that use IIOP and send JPA entities from the server to the client will fail during deserialization if the entities are Serializable (but not Externalizable) and do not declare a `writeObject()` method.

Workaround

Add a `writeObject()` method to such entity classes. The write object can be trivial:

```
private void
writeObject(java.io.ObjectOutputStream out)
   throws IOException {
  out.defaultWriteObject();
}
```

2.10.12 Non-Transactional Message-Driven Bean Container Can Fail to Provide Reproducible Behavior For Foreign Topics

Platform: All

When using multi-threaded processing for non-transactional topic Message-Driven Beans (MDBs) that specify a foreign topic (non-WebLogic) JMS, the MDB container can fail to provide reproducible behavior. For example, if a `runtimeException` is thrown in the `onmessage()` method, the container may still acknowledge the message.

Workaround

Set the `max-beans-in-free-pool` attribute to 1 in the deployment descriptor.

2.11 Examples Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.11.1, "Security Configuration in medrec.wls.config"
- Section 2.11.2, "HTML File not Created for StreamParser.java File"
- Section 2.11.3, "Warning Message Appears When Starting Medrec or Samples Domain"

2.11.1 Security Configuration in medrec.wls.config

Platform: All

The `medrec.wls.config` target in *SAMPLES_HOME*`/server/medrec/setup/build.xml` has a known issue with respect to security configuration.

2.11.2 HTML File not Created for StreamParser.java File

Platform: All

The ../xml/stax example contains two files with the same root but different extensions: StreamParser.java and StreamParser.jsp. The samples viewer build, however, creates just one corresponding HTML file, rather than two for each type of file. In this case only the StreamParser.jsp file has an equivalent HTML file; the StreamParser.java file does not.

The problem occurs because of a setting in the build.xml file that controls the behavior of java2html to generate the files for the documentation.

When using java2html, the useShortFileName="true" parameter crops off the file extensions for the source files to create the file names for the HTML output files. If two files have the same name and different file extensions, whichever HTML file is generated last will overwrite previous ones.

Workaround

Set the useShortFileName parameter to "false". This setting generates HTML files with the file extensions included in the name. The drawback to this solution is that every link that points to the HTML output file needs to be revised, regardless of whether the files in question were affected by the bug.

2.11.3 Warning Message Appears When Starting Medrec or Samples Domain

Platform: All

When you start the medrec or samples domains, you may see a warning message similar to this:

```
<Warning> <WorkManager> <BEA-002919> <Unable to find a WorkManager with name
weblogic.wsee.mdb.DispatchPolicy. Dispatch policy
weblogic.wsee.mdb.DispatchPolicy will map to the default WorkManager for the
application bea_wls_async_response>
```

This warning message appears in the standard output of the Console while starting a WebLogic Server sample application with an asynchronous Web Service deployed.

Workaround

The warning is harmless and can be ignored.

2.12 HTTP Publish/Subscribe Server Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.12.1, "Authentication and Authorization of the Local Client is not Supported"
- Section 2.13.2, "Improper Rollback to Previous Installation May Occur After Exiting an Upgrade Installation Prematurely"
- Section 2.13.3, "WebLogic Server Installer Fails With Insufficient Disk Space Error"

2.12.1 Authentication and Authorization of the Local Client is not Supported

Platform: All

The HTTP Publish/Subscribe server does not support authentication and authorization of the local client. The local client has full permissions to operate on channels of the HTTP Publish/Subscribe server, which means the local client can create/delete channels and publish/subscribe events from channels.

2.12.2 Event Messages Published by Local Clients Cannot Be Received

Platform: All

In a clustering environment, event messages published by a local client on a server can be received only by subscribed clients connected to the same server. These messages cannot be received by subscribed clients connected to other servers in the cluster.

2.12.3 Event Messages Published By Local Clients Do Not Go Through Filters

Platform: All

Event messages published to a channel by a local client will not go through the Message Filters configured to that channel.

2.13 Installation Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.13.1, "Sybase JDBC Drivers Not Downloaded with Upgrade Installation"
- Section 2.13.2, "Improper Rollback to Previous Installation May Occur After Exiting an Upgrade Installation Prematurely"
- Section 2.13.3, "WebLogic Server Installer Fails With Insufficient Disk Space Error"
- Section 2.13.4, "WebLogic Server Home Cannot Be the Same as the Middleware Home"
- Section 2.13.5, "Installation Fails with Fatal Error"
- Section 2.13.6, "WebLogic Server Installation Fails When Installed Using Network Drive"
- Section 2.13.7, "Exception When Using WLS 10.3.6 With Sybase 15.7"

2.13.1 Sybase JDBC Drivers Not Downloaded with Upgrade Installation

Platform: All

The Oracle WebLogic Server 11*g* Release 1 installer does not download the Sybase JDBC drivers. When you try to upgrade an existing WebLogic Server 10.3 installation using the latest installer, it does not remove the Sybase JAR files from the original installation. The installer upgrades only the `weblogic.jar` file.

The Sybase JAR files (jconn2.jar, jconn3.jar, and jConnect.jar) in the /server/lib or /server/ext/jdbc/sybase directories are removed from the manifest classpath in the upgraded weblogic.jar file. Therefore, if the classpath of a WebLogic Server application does not include Sybase JAR files and only includes weblogic.jar then after the upgrade installation, the application will throw a ClassNotFoundException.

To work around this issue, explicitly add Sybase JAR files in the WebLogic Server application classpath.

2.13.2 Improper Rollback to Previous Installation May Occur After Exiting an Upgrade Installation Prematurely

Platform: All

When using an Upgrade installer or Smart Update to upgrade an existing WebLogic Server 10.3.x installation to WebLogic Server 10.3.4, if you abort the upgrade before completion, the installation should automatically roll back to the prior installation. This may not always occur, resulting in an unusable installation.

2.13.3 WebLogic Server Installer Fails With Insufficient Disk Space Error

Platform: All

The WebLogic Server installer can fail with an insufficient disk space error, even when there is a large amount of available disk space on the file system or disk.

Workaround

Use the `-Dspace.detection` property in the installation command to disable the available space check. For example:

```
java -Xmx1024M -Dspace.detection=false -jar installer_file_name
-mode=silent -silent_xml=silent.xml
```

or

```
wls1034_linux.bin -Dspace.detection=false
```

2.13.4 WebLogic Server Home Cannot Be the Same as the Middleware Home

Platform: MS Windows

You cannot use the Middleware home directory as the WebLogic Server home directory. For example, if the Middleware home directory is C:\Oracle\Middleware, you cannot specify C:\Oracle\Middleware as the home directory for WebLogic Server. Doing so has the potential to cause serious issues with the Configuration Wizard, Template Builder, and possibly other WebLogic Server tools.

Workaround

Install WebLogic Server in a directory other than the Middleware home directory. For example, if the Middleware home directory is C:\Oracle\Middleware, it is permissible to install WebLogic Server in C:\Oracle\Middleware\wlserver or C:\Oracle\wlserver.

2.13.5 Installation Fails with Fatal Error

Platform: All UNIX

The installer does not verify whether sufficient disk space is available on the machine prior to completing the installation. As a result, if an installation cannot be completed due to insufficient space, the installer displays the following error message and exits:

```
Fatal error encountered during file installation. The installer will now
cleanup and exit!
```

Workaround

If this problem occurs, restart the installer using the following command:

```
server103_linux32.bin -log=log.out -log_priority=debug
```

The preceding command generates a log of the installation procedure, providing details about the exact cause of the failure. If the cause is indeed insufficient space, the log file indicates it explicitly.

2.13.6 WebLogic Server Installation Fails When Installed Using Network Drive

Platform: MS Windows

The installation of WebLogic Server would fail when the installer is accessed from another machine in the network throwing an error message:

```
com.bea.plateng.wizard.installer.utils.SelfExtract - Error accessing jar
```

Workaround

Perform either of the following:

1. Map the installer location that is present in the network machine to the local machine drive and then install.

   Steps to map the drive are:

   a. Right click on **My Computer** and select the **Map Network Drive** option.

   b. Select a drive and browse and select the Network path of the machine where the installer is present.

   c. Click on **Finish**.

   Then the installation could be done using the command `java -XX:MaxPermSize=1024m -jar Y:\wls1031_generic.jar` where `Y:` is the local mapped drive.

2. Copy the WLS installer from the network drive to the local machine folder and point it to the location while installing.

   For example:

   ```
   java -XX:MaxPermSize=1024m -jar c:\Installer\wls1031_generic.jar
   ```

2.13.7 Exception When Using WLS 10.3.6 With Sybase 15.7

When using WLS 10.3.6 with Sybase 15.7, you may get the following exception when processing binary data:

```
[Sybase JDBC Driver][Sybase]The token datastream length was not correct.
This is an internal protocol error.
```

This is a limitation of the Sybase driver that is packaged with WLS 10.3.6 as against Sybase 15.7.

Workaround

Contact My Oracle Support and ask for the latest WLS driver for Sybase.

2.14 Java EE Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.14.1, "FastSwap May Relax the Access Modifiers of Fields and Methods"
- Section 2.14.2, "FastSwap Does Not Support Redefinition of the Entity Bean and ejbClass"
- Section 2.14.3, "Classpath Order Is Not Guaranteed When There Are Multiple JARs in an EAR File"

2.14.1 FastSwap May Relax the Access Modifiers of Fields and Methods

Platform: All

FastSwap may relax the access modifiers of fields and methods. Private and protected members may be made public at runtime. This changes the behavior of reflection and may affect reflection-based frameworks such as Struts.

2.14.2 FastSwap Does Not Support Redefinition of the Entity Bean and ejbClass

Platform: All

FastSwap does not support redefinition of the Entity bean and ejbClass (Session/MDB). Therefore, any updates to entity classes will cause redefinition errors.

Workaround

After updating an entity class, redeploy the application.

2.14.3 Classpath Order Is Not Guaranteed When There Are Multiple JARs in an EAR File

Platform: All

When you have an EAR file containing separate JAR files, and two or more of those JAR files have a class with the same name, it is not possible to predict from which of those JAR files WebLogic Server will instantiate the class. This is not an issue if the classes are the same, but if they are different implementations, the results are unpredictable.

Workaround

Currently there is no known workaround for this issue.

2.15 JDBC Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.15.1, "Call To setTransactionIsolation() May Fail When Using the JDBC Driver for MS SQLServer"
- Section 2.15.2, "An Attempt to Access a Remote 10.3.2 or Later WLS Data Source Fails"
- Section 2.15.3, "ORA-01591 Errors Occur on SOA Servers Configured to Use Multiple Oracle RAC Nodes"

2.15.1 Call To setTransactionIsolation() May Fail When Using the JDBC Driver for MS SQLServer

Platform: All

When using the JDBC driver for MS SQLServer, a call to `setTransactionIsolation()` may fail in a transactional context if `getTransactionIsolation()` is called first.

2.15.2 An Attempt to Access a Remote 10.3.2 or Later WLS Data Source Fails

Platform: All

A new system property, `-Dweblogic.jdbc.remoteEnabled`, has been added to JDBC in Oracle WebLogic Server 10.3.2. For compatibility with prior releases of WebLogic Server, the default setting of this property is `true`. When this property is set to `false`, remote JDBC access is turned off, and such access results in an exception.

Remote access may occur explicitly in an application, or implicitly during a global (XA/JTA) transaction with a participating non-XA data source that is configured with the LLR, 1PC or Emulate XA global transaction option. The following enumerates the cases when an exception will be thrown, and work-arounds for each case (if any).

An exception occurs in the following cases. A workaround (if any) for a given case is provided.

- When a stand-alone client application uses any type of data source.
- When an application that is hosted on WebLogic Server uses any type of data source, and the data source is not configured (targeted) locally. A potential workaround is to target the data source locally.
- When accessing a same named non-XA data source with a transaction option of LLR, 1PC or Emulate XA on multiple WebLogic Server instances in the same global transaction. In this case, there are two potential work-arounds:
  - Change data sources to use XA instead (this may lower performance), or
  - For the 1PC/emulateXA types, change the application to ensure the data source is accessed from a single server.
- When accessing a non-XA data source with the LLR transaction option on a server that is different than the transaction coordinator. For server-initiated transactions, the coordinator location is chosen based on the first participating resource in the transaction. In this case, there are two potential work-arounds: (a) change the data source to use XA instead (this may lower performance); or (b) change the application to ensure data source access on the transaction coordinator, as described in "Optimizing Performance with LLR" in Oracle Fusion Middleware Programming JTA for OracleWebLogic Server. The latter may not be possible in some cases; for example, when an MDB application receives messages from a remote WebLogic JMS server, the transaction coordinator will always be the WebLogic server that's hosting the JMS server, but it may not be possible to move the MDB application to the same WebLogic server.
  - Change the data source to use XA instead (this may lower performance), or
  - Change the application to ensure data source access on the transaction coordinator, as described in "Optimizing Performance with LLR" in *Programming JTA for Oracle WebLogic Server*. This workaround may not be possible in some cases. For example, when an MDB application receives messages from a remote WebLogic JMS server, the transaction coordinator will always be the WebLogic Server instance that is hosting the JMS server, but it may not be possible to move the MDB application to the same WebLogic Server instance.

2.15.3 ORA-01591 Errors Occur on SOA Servers Configured to Use Multiple Oracle RAC Nodes

Platform: All

On SOA servers using multiple Oracle RAC database nodes, when WebLogic Server multi data sources are configured for XA and load balancing, ORA-10591 errors can occur.

Workaround

Download and apply Oracle RAC database patch 7675269 for Linux x86, Oracle Release 11.1.0.7.0. You can download this patch from My Oracle Support. Alternatively, you can download and apply patch set 9007079 for Linux x86, Oracle Release 11.1.0.7.0, which includes the patch 7675269.

2.16 JMS Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.16.1, "Deployment Descriptor Validation Fails"
- Section 2.16.2, "Exception When Multiple Producers Use the Same Client SAF Instance"
- Section 2.16.3, "Multi-byte Characters are not Supported in Store File and Directory Names"
- Section 2.16.4, "Generation of the Default UOO Name Has Changed"
- Section 2.16.5, "Testing Abrupt Failures of WebLogic Server When Using File Stores on NFS"
- Section 2.16.6, "JMS Message Consumers Will Not Always Reconnect After a Service Migration"
- Section 2.16.7, "Forcing Unicast Messages To Be Processed in Order"
- Section 2.16.8, "JMS Distributed Destinations Are Not Present After Extending a Domain"

2.16.1 Deployment Descriptor Validation Fails

Platform: All

Deployment descriptor validation fails when descriptor validation is enabled, and an EAR file contains only JMS modules.

Workaround

Make sure that there is at least one Java EE specification-compliant module in the EAR.

2.16.2 Exception When Multiple Producers Use the Same Client SAF Instance

Platform: All

When multiple JMS producers use the same JMS Client SAF instance (within a single JVM), depending on the timing of the JMS SAF client creation, you might receive the following exception:

```
Error getting GXA resource [Root exception is weblogic.jms.common.JMSException:
weblogic.messaging.kernel.KernelException: Error getting GXA resource]
```

Workaround

When using multiple JMS SAF client producers, try introducing a small delay between the creation of each new client.

2.16.3 Multi-byte Characters are not Supported in Store File and Directory Names

Platform: All

There is no support for multi-byte characters in WebLogic Store file and directory names. For instance, when the WebLogic Server name has multi-byte characters, the default store cannot be created, and WebLogic Server will not boot.

Workaround

Create WebLogic Server instances without multi-byte characters in the path name and use that path name for the default store configuration. Do not use multi-byte characters in the Weblogic Server name.

2.16.4 Generation of the Default UOO Name Has Changed

Platform: All

WebLogic Server 10.3.4 contains a fix for configurations that set a default unit-of-order (UOO) on a JMS regular destination, distributed destination, or template. This fix ensures that the default unit-of-order name stays the same even after a restart of the destination's host JMS server. The default UOO name is now based on the domain, JMS server, and destination names.

2.16.5 Testing Abrupt Failures of WebLogic Server When Using File Stores on NFS

Platform: All

Oracle strongly recommends verifying the behavior of a server restart after abrupt machine failures when the JMS messages and transaction logs are stored on an NFS mounted directory. Depending on the NFS implementation, different issues can arise post failover/restart. For more information, see Section 6.3, "Testing Abrupt Failures of WebLogic Server When Using File Stores on NFS."

2.16.6 JMS Message Consumers Will Not Always Reconnect After a Service Migration

Platform: All

JMS message consumers will not always reconnect after a service migration when an application's `WLConnection.getReconnectPolicy()` attribute is set to `all`. If the consumers do not get migrated, either an exception is thrown or `onException` will occur to inform the application that the consumer is no longer valid.

Workaround

The application can refresh the consumer either in the exception handler or through `onException`.

2.16.7 Forcing Unicast Messages To Be Processed in Order

Platform: All

Certain conditions can cause very frequent JNDI updates, and as a result, JMS subscribers may encounter a `java.naming.NameNotFoundException`. For more information, see Section 2.8.5, "Forcing Unicast Messages To Be Processed in Order."

2.16.8 JMS Distributed Destinations Are Not Present After Extending a Domain

Platform: All

After extending a domain using an extension template that was generated from a domain that contains JMS distributed destinations, the distributed destinations are not present in the domain. This impacts the following distributed destinations:

- *distributed-queue*
- *distributed-topic*
- *uniform-distributed-queue*
- *uniform-distributed-topic*

If any of these elements are contained in the JMS XML files in the source template, they are not processed and are not configured in the destination domain.

Workaround

To resolve this, use the following sequence of WLST commands, either interactively or in a script:

```
readDomain('domain_path')

addTemplate('extension_template_file')

unassign('JmsSystemResource','resource_name','Target','destination_name')
For example: unassign('JmsSystemResource','JMSModule','Target','C1’)

assign('JmsSystemResource','resource_name','Target','destination_name')
For example: assign('JmsSystemResource','testModule','Target','Server-1')

unassign('JmsSystemResource','resource_name','Target','destination_name')
For example: unassign('JmsSystemResource','testModule','Target','Server-1')

assign('JmsSystemResource','resource_name','Target','destination_name')
For example: assign('JmsSystemResource','testModule','Target','C1')

updateDomain()

closeDomain()
```

2.17 JNDI Issues and Workarounds

There are no known JNDI issues in this release of WebLogic Server.

2.18 JSP and Servlet Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.18.1, "Deployment Plans Cannot Be Used To Override Two Descriptors"
- Section 2.18.2, "Spring Dependency Injection Not Supported on JSP Tag Handlers"
- Section 2.18.3, "503 Error When Accessing an Application With a Valid sessionid"
- Section 2.18.4, "Java SE 7 Language Enhancements Are Not Supported Within JSP Pages"

2.18.1 Deployment Plans Cannot Be Used To Override Two Descriptors

Platform: All

Deployment plans cannot be used to override the following two descriptors during deployment of a Web application or a Web module: WEB-INF/classes/META-INF/persistence.xml and WEB-INF/classes/META-INF/persistence-configuration.xml. Deployment plans can otherwise be used to override any descriptor.

Workaround

Package WEB-INF/classes/META-INF/persistence.xml and WEB-INF/classes/META-INF/persistence-configuration.xml (if present) along with related class files into a JAR file. The JAR file must then be placed in the WEB-INF/lib directory of the Web application or Web module. A deployment plan can be used to override the two descriptors in such a JAR file.

2.18.2 Spring Dependency Injection Not Supported on JSP Tag Handlers

Platform: All

With the Spring extension model enabled, WebLogic Server 10.3 or later does not support Spring Dependency Injection (DI) on JSP tag handlers for performance reasons.

Currently, WebLogic Server supports Spring DI on most Web components, for example, servlets, filters and listeners. Spring DI is not, however, presently supported on JSP tag handlers for performance reasons.

2.18.3 503 Error When Accessing an Application With a Valid sessionid

Platform: All

When a session is persistent and an older version of a servlet context is retired, accessing the application with a valid `sessionid` will cause a 503 error.

For example, the session-persistent type of a versioned Web application is 'file'. A user can access the application successfully. Later, version 2 of the application is redeployed and version 1 is retired. If the same user accesses the application, they will get a 503 error.

2.18.4 Java SE 7 Language Enhancements Are Not Supported Within JSP Pages

Platform: All

Java SE 7 introduced new programming language enhancements, as listed here: `http://docs.oracle.com/javase/7/docs/technotes/guides/language/enhancements.html`.

This release of WebLogic Server does not support the use of the new Java SE 7 enhancements within any aspect of a JSP page. Note that this does not prevent you from using a Java SE 7 runtime to execute WebLogic Server and run JSP pages that use Java SE 6 language syntax and below.

Workaround

If JSP pages use any of the new programming language enhancements introduced in Java SE 7, upgrade to WebLogic Server 12.1.3.

2.19 JTA Issues and Workarounds

There are no known JTA issues in this release of WebLogic Server.

2.20 Java Virtual Machine (JVM) Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.20.1, "1.4 Thin Client Applet Cannot Contact WebLogic Server"

- Section 2.20.2, "Applications Running on Some Processors May Experience Intermittent Time Issues"
- Section 2.20.3, "JRockit JVM Appears to Freeze When Doing Long Array Copies"
- Section 2.20.4, "Serial Version UID Mismatch"
- Section 2.20.5, "JVM Stack Overflow"
- Section 2.20.6, "Using AWT libraries May Cause a JVM Crash"

2.20.1 1.4 Thin Client Applet Cannot Contact WebLogic Server

Platform: All

Due to a known Sun Microsystems VM bug (513552), a 1.4 Thin Client Applet cannot contact WebLogic Server 9.0 or later. This is because the VM does not distinguish correctly between a client and a server connection. The VM creates a server-type connection and caches it. It then attempts to make a client-type connection, finds the cached connection and tries to use that, but then encounters an error because clients are not allowed to use server connections.

2.20.2 Applications Running on Some Processors May Experience Intermittent Time Issues

Platform: RedHat Linux

Applications that run on RH Linux on Intel G5 processors and that also directly or indirectly use system time calls may experience intermittent time issues if the `ClockSource` is set to `tsc` (the default). The standard POSIX C `gettimeofday()` call, and consequently also the Java `System.currentTimeMillis()` and `java.util.Date()` calls can intermittently return a value that is approximately 4400 seconds in the future, even in a single-threaded application.

This issue is not unique to WebLogic or Java, but applies to any application running on RH Linux on Intel G5 processors. Issues can occur for applications that either explicitly make a time call using standard Java, or explicitly by using any time-based application server services.

Possible symptoms include, but are not limited to, premature transaction timeouts, unexpected expiration of JMS messages, and incorrectly scheduled timers.

If you're interested in a standalone reproducer for this problem, contact Oracle and reference bug number 8160147.

Workaround

There is no known official patch for Linux. Instead, change the clock source from `tsc` to `hpet`. After making this modification on test systems, exceptions due to invalid `System.currentTimeMillis()/gettimeofday()` return values were no longer seen. To change the system clock from `tsc` to `hpet` on a trial basis, perform the following steps as `root`:

1. Disable ntpd (if running)
2. Echo 'hpet' &gt; /sys/devices/system/clocksource/clocksource0/current_clocksource
3. Enable ntpd

Note that this change will not survive a reboot. For more information, please see: `http://www.gossamer-threads.com/lists/linux/kernel/813344`

2.20.3 JRockit JVM Appears to Freeze When Doing Long Array Copies

Platform: Linux

The JRockit JVM appears to freeze when doing long array copies as part of unlimited forward rolling. This can happen when multiple server reboots occur due to `Out Of Memory` conditions.

Workaround

When booting the servers, include the following JRockit JVM flag:

```
 -XXrollforwardretrylimit:-1
```

2.20.4 Serial Version UID Mismatch

Platform: Linux

A Serial Version UID Mismatch issue is encountered if you deploy an application on a latest JVM, but compiled with previous Service Release of IBM Java 6 JDK.

Workaround

To be compatible with the serialization of previously compiled applications, modify the *`BEA_HOME`*`/wlserver_10.3/common/bin/commEnv.sh` file to include the following command:

```
JAVA_OPTIONS="$JAVA_OPTIONS
-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"
```

Alternatively, you can use the command line option:

```
export IBM_JAVA_OPTIONS=
"-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"
```

If you intend to deploy new applications with previously compiled applications, they must be recompiled as necessary to have the same Serial Version UID.

2.20.5 JVM Stack Overflow

Platform: Linux

You might encounter a JVM stack overflow error or exception while running WebLogic Server. This issue applies to Oracle Enterprise Linux 4, 5, 5.1 on AMD64 and 64-bit Xeon platforms.

Workaround

Increase the stack size from the default 128k to 256k.

2.20.6 Using AWT libraries May Cause a JVM Crash

Platform: Linux x86

You might encounter a JVM crash when using GUI libraries such as AWT or javax.swing (which often delegates to AWT).

Workaround

Start the server using the following flag:

```
-Djava.awt.headless=true
```

2.21 Monitoring Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.21.1, "MBean Attributes Not Explicitly Marked as @unharvestable Appear as Harvestable"
- Section 2.21.2, "Events Generated By the JVM Level Are Not Generated at Low Volume"
- Section 2.21.3, "WLDF Performance Issues Can Occur When JVM Events Are Enabled"

2.21.1 MBean Attributes Not Explicitly Marked as @unharvestable Appear as Harvestable

Platform: All

The `@unharvestable` tag is not being honored at the interface level. If MBean attributes are not explicitly marked as `@unharvestable`, they are considered to be harvestable and will appear as harvestable in the WebLogic Administration Console.

Workaround

You can explicitly mark MBean attributes as `@unharvestable`.

2.21.2 Events Generated By the JVM Level Are Not Generated at Low Volume

Platform: All

In WebLogic Server 10.3.3, the default WLDF diagnostic volume setting was Off. As of WebLogic Server 10.3.4, the default diagnostic volume setting is Low Volume, and events generated by the JVM level are not being generated at the Low Volume setting in WebLogic Server 10.3.4 (JVM-level events were generated at the Low Volume setting in WebLogic Server 10.3.3). The JVM-level events are still generated at the High Volume and Medium Volume settings in WebLogic Server 10.3.4.

Workaround

Use one of the following workarounds to cause the JVM-level events to be generated:

- Increase the WLDF diagnostic volume to the Medium or High level.
- Use JRMC, JRCMD, or the JRockit command line settings to activate a separate flight recording in the WebLogic Server instance. By doing so, JVM will cause JVM events to be present at all WLDF diagnostic volume settings (Off, Low, Medium, and High).

2.21.3 WLDF Performance Issues Can Occur When JVM Events Are Enabled

Platform: All

When JVM events are enabled, WLDF performances issues may occur in the following situations:

- If there are no other JRockit flight recordings enabled, performance can degrade when the WLDF diagnostic volume is set to Medium or High level.
- If other JRockit flight recordings are enabled, performance can degrade at all WLDF diagnostic volume levels (Off, Low, Medium, and High).

2.22 Node Manager Issues and Workarounds

This section describes the following Node Manager issues and workarounds:

- Section 2.22.1, "Node Manager Causes Uncaught SSLException During Managed Server Startup with JSSE"

2.22.1 Node Manager Causes Uncaught SSLException During Managed Server Startup with JSSE

Platform: All

When Node Manager and the Administration Server are running with JSSE enabled and you start Managed Servers, the following uncaught exceptions from Node Manager occur:

```
<Jun 11, 2014 4:50:53 AM> <WARNING> <Uncaught exception in server
handlerjavax.net.ssl.SSLException: Inbound closed before receiving peer's close_
notify: possible truncation attack?>
```

These errors are harmless and can be ignored.

2.23 Operations, Administration, and Management Issues and Workarounds

There are no known Operations, Administration, and Management issues in this release of WebLogic Server.

2.24 Oracle Kodo Issues and Workarounds

There are no known Oracle Kodo issues in this release of WebLogic Server.

This section describes the following Oracle Kodo issue and workaround:

- Section 2.24.1, "Value Retrieved for an Empty Byte Array Field is NULL"

2.24.1 Value Retrieved for an Empty Byte Array Field is NULL

Platform: MS Windows 2000

When trying to persist an empty byte array field within an entity to a Sybase or Oracle database, the value gets stored as a NULL rather than as bytes. As a result, when retrieving the value, NULL is returned.

This is a limitation of the Sybase and Oracle drivers, which convert the empty byte array to a NULL while storing it in the database. The issue happens with Weblogic JDBC drivers as well as the proprietary Sybase and Oracle drivers.

2.25 Plug-ins Issues and Workarounds

This section describes the following issues for various WebLogic Server plug-ins:

- Section 2.25.1, "apr_socket_connection Exception Occurs When Using the IIS Plug-In"

2.25.1 apr_socket_connection Exception Occurs When Using the IIS Plug-In

Platform: All

Under the following circumstances, the IIS plug-in may not work, resulting in an `apr_socket_connection` error:

1. Both the IIS and Weblogic Server instances are on the same machine.
2. IPv6 is enabled on the machine, but the machine is not in an IPv6 environment (that is, the IPv6 interface is enabled but is not working).
3. The listen address of the WebLogic Server instance is set to the simple host name.
4. Either the directive WebLogicHost or WebLogicCluster is set to the simple host name for the IIS instance.

2.26 Protocols Issues and Workarounds

There are no known Protocols issues in this release of WebLogic Server.

2.27 RMI-IIOP Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.27.1, "Ant 1.7 rmic Task Incompatibility"

2.27.1 Ant 1.7 rmic Task Incompatibility

Platform: All

Calls to the Ant version 1.7 `rmic` task automatically add a `-vcompat flag`, which is not compatible with `rmic` for Oracle WebLogic Server.

Workaround

Use either of the following workarounds if your rmic call is of the form:

```
rmic classname="com.bea.crmsimulation.legacyra.LegacyAdapter"
   base="${module_location}/core-legacy-ra/classes"
   classpath="${core.classes}" compiler="weblogic" />
```

- Add a `stubversion`

  ```
  <rmic classname="com.bea.crmsimulation.legacyra.LegacyAdapter"
     base="${module_location}/core-legacy-ra/classes"
     classpath="${core.classes}" compiler="weblogic"
     stubversion="1.2"/>
  ```

- Remove the `compiler` flag

  ```
  <rmic classname="com.bea.crmsimulation.legacyra.LegacyAdapter"
     base="${module_location}/core-legacy-ra/classes"
     classpath="${core.classes}"
  ```

2.28 Security Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.28.1, "StoreBootIdentity Works Only if the Appropriate Server Security Directory Exists"
- Section 2.28.2, "Boot Time Failure Occurs With SecurityServiceException"
- Section 2.28.3, "Authentication Failure After Upgrading a Domain From WLS 6.1"

2.28.1 StoreBootIdentity Works Only if the Appropriate Server Security Directory Exists

Platform: All

The option `-Dweblogic.system.StoreBootIdentity` works only if the appropriate server security directory exists. This directory is usually created by the Configuration Wizard or upgrade tool.

However, the appropriate server security directory could be absent in domains checked into source-control systems.

2.28.2 Boot Time Failure Occurs With SecurityServiceException

Platform: All

A WebLogic Server instance can experience a boot time failure with a `SecurityServiceException` when the RDBMS Security Data Store is configured for a DB2 database using the DB2 driver supplied with WebLogic Server.

Workaround

When RDBMS Security Data Store is using the `AlternateId` connection property for a DB2 database, you must also set the additional property `BatchPerformanceWorkaround` as `true` when using the DB2 driver supplied with WebLogic Server.

2.28.3 Authentication Failure After Upgrading a Domain From WLS 6.1

Platform: All

After upgrading a domain from WLS 6.1, the WebLogic Server instance will not boot due to an authentication failure.

Workaround

A system user password must be set up in the WLS 6.1 domain before or after the upgrade process in order for the WebLogic Server instance to boot properly.

2.28.4 InvalidParameterException Message Generated and Displayed

Platform: All

After you configure either the Identity Provider or Service Provider services for SAML 2.0 and attempt to publish the SAML 2.0 services metadata file, an `InvalidParameterException` message may be generated and displayed in the Administration Console.

Workaround

When configuring the SAML 2.0 federation services for a WebLogic Server instance, be sure to enable all binding types that are available for the SAML role being configured. For example, when configuring SAML 2.0 Identity Provider services, you should enable the POST, Redirect, and Artifact bindings. When configuring SAML 2.0 Service Provider services, enable the POST and Artifact bindings. Optionally, you may choose a preferred binding.

2.28.5 Enabling Both the Authentication and Passive Attributes In SML 2.0 Service Provider Services Is an Invalid Configuration

Platform: All

When configuring SAML 2.0 Service Provider services, enabling both the Force Authentication and Passive attributes is an invalid configuration that WebLogic Server is unable to detect. If both these attributes are enabled, and an unauthenticated user attempts to access a resource that is hosted at the Service Provider site, an exception is generated and the single sign-on session fails.

Note that the Force Authentication attribute has no effect because SAML logout is not supported in WebLogic Server. So even if the user is already authenticated at the Identity Provider site and Force Authentication is enabled, the user is not forced to authenticate again at the Identity Provider site.

Avoid enabling both these attributes.

2.28.6 Running the WebLogic Full Client in a Non-Forked VM

Platform: All

If the WebLogic Full Client is running in a non-forked VM, for example by means of a `<java>` task invoked from an Ant script without the `fork=true` attribute, the following error might be generated:

```
java.lang.SecurityException: The provider self-integrity check failed.
```

This error is caused by the self-integrity check that is automatically performed when the RSA Crypto-J library is loaded. (The Crypto-J library, `cryptoj.jar`, is in the `wlfullclient.jar` manifest classpath.)

This self-integrity check failure occurs when the client is started in a non-forked VM and it uses the Crypto-J API, either directly or indirectly, as in the following situations:

- The client invokes the Crypto-J library directly.
- The client attempts to make a T3S connection, which triggers the underlying client SSL implementation to invoke the Crypto-J API.

When the self-integrity check fails, further invocations of the Crypto-J API fail.

Workaround

When running the full client in a `<java>` task that is invoked from an Ant script, always set the `fork` attribute to `true`.

For more information about the self-integrity check, see "How a Provider Can Do Self-Integrity Checking" in How to Implement a Provider in the Java™ Cryptography Architecture, available at the following URL:

`http://download.oracle.com/javase/6/docs/technotes/guides/security/crypto/HowToImplAProvider.html#integritycheck`

2.28.7 Random Number Generator May Be Slow on Machines With Inadequate Entropy

Platform: Linux

In order to generate random numbers that are not predictable, SSL security code relies upon "entropy" on a machine. Entropy is activity such as mouse movement, disk IO, or network traffic. If entropy is minimal or non-existent, then the random number generator will be slow, and security operations may time out. This may disrupt activities such as booting a Managed Server into a domain using a secure admin channel. This issue generally occurs for a period after startup. Once sufficient entropy has been achieved on a JVM, the random number generator should be satisfied for the lifetime of the machine.

For further information, see Sun bugs 6202721 and 6521844 at:

`http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6202721`

`http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6521844`

Workaround

On low-entropy systems, you can use a non-blocking random number generator, providing your site can tolerate lessened security. To do this, add the `-Djava.security.egd=file:///dev/urandom` switch or `file:///dev/./urandom` to the command that starts the Java process. Note that this workaround should not be used in production environments because it uses pseudo-random numbers instead of genuine random numbers.

2.28.8 WebLogic Server Does Not Support Server Name Indication

Although JSSE supports Server Name Indication (SNI) in its SSL implementation, WebLogic Server does not support SNI.

2.28.9 Using SHA256 with WebLogic Server 10.3.6

Platform: All

When you try to load a certificate chain using utils.ImportPrivateKey or utils.ValidateCertChain, the SHA256 certificate authority fails to load.

This is because certificate loading utilities such as, ImportPrivateKey and ValidateCertChain use Certicom, and Certicom does not support SHA256 CA.

Workaround

If you want to use SHA256 with WLS 10.3.6 then you need to enable JSSE and use the `-Dweblogic.security.SSL.enableJSSE=true` flag with all certificate and keystore utilities provided by WebLogic Server.

2.29 SNMP Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.29.1, "Cannot Use SNMP to Send Traps Based on JMX Monitors for Non-Integer MBean Attributes"

2.29.1 Cannot Use SNMP to Send Traps Based on JMX Monitors for Non-Integer MBean Attributes

Platform: All

The SNMP gauge and counter monitor configuration only accepts integer values as thresholds. As a result, these monitors cannot be configured to send traps based on MBean attributes that are non-integer types such as `float`, `long` or `double`. The corresponding JMX-based gauge and counter monitor supports a threshold value which is a `java.lang.Number`. However, the current SNMP implementation in WebLogic Server accepts only an `int` value.

Workaround

You can use the Watch and Notification component of the WebLogic Diagnostic Framework (WLDF) to monitor any MBean attribute and generate a trap based on WLDF watch rules. A watch rule can be configured to monitor the thresholds of MBean attributes and send a SNMP notification when the watch rule evaluates to `true`.

For more information, see "Configuring Watches and Notifications" in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

2.30 Spring Framework on WebLogic Server Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.30.1, "OpenJPA ClassFileTranformer Does Not Work When Running on JRockit"
- Section 2.30.2, "petclinic.ear Does Not Deploy on WebLogic Server"

2.30.1 OpenJPA ClassFileTranformer Does Not Work When Running on JRockit

Platform: All

The OpenJPA `ClassFileTranformer` does not work when running WebLogic Server on JRockit.

Workaround

Use an alternative method of applying enhancements at build time through an OpenJPA enhancer compiler; do not use the `LoadTimeWeaver`.

2.30.2 petclinic.ear Does Not Deploy on WebLogic Server

Platform: All

For the SpringSource `petclinic` sample, the `petclinic.war` deploys without any problems. The `petclinic.ear` will not deploy on WebLogic Server because it is not packaged correctly. A request has been sent to SpringSource to fix the `petclinic.ear` packaging.

2.31 System Component Architecture (SCA) Issues and Workarounds

There are no known SCA issues in this release of WebLogic Server.

2.32 Upgrade Issues and Workarounds

This section describes the following issue:

- Section 2.32.1, "Domains Created on WebLogic Server 10.3.1 Cannot Be Run on WebLogic Server 10.3"

2.32.1 Domains Created on WebLogic Server 10.3.1 Cannot Be Run on WebLogic Server 10.3

Platform: All

If you create a domain using WebLogic Server 10.3.1, then roll back to WebLogic Server 10.3, you will not be able to start the servers that you created in that domain. This is a known restriction, as the `config.xml` file contains references to newer schema definitions (`xmlns.oracle.com`) that did not exist in WebLogic Server 10.3.

2.33 Web Applications Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.33.1, "Administration Console Fails to Implement session-timeout Changes"
- Section 2.33.2, "Connection Pool Connection Reserve Timeout Seconds Value is Overridden"
- Section 2.33.3, "Database Connections Become Unstable When a PoolLimitSQLException Occurs"
- Section 2.33.4, "Web Page Fails to Open When Accessing It Using the SSL Port"
- Section 2.33.5, "Unable to View the Output of a JSPX Page in Internet Explorer"
- Section 2.33.6, "Unable to View the Output of SVG files in Internet Explorer 7"
- Section 2.33.7, "Changed the Expires Format For the Cookie Date in Version 0"

2.33.1 Administration Console Fails to Implement session-timeout Changes

Platform: All

If the `session-timeout` is configured in the `web.xml` file, any changes made to change the `session-timeout` using the Administration Console do not take effect.

Workaround

Use a deployment plan to override the `session-timeout` setting.

2.33.2 Connection Pool Connection Reserve Timeout Seconds Value is Overridden

Platform: All

When using a JDBC session, the value of Connection Reserve Timeout Seconds for a connection pool is changed to be one of the following:

- the JDBC connection timeout seconds, which is defined in the session descriptor (either in `weblogic.xml` or `weblogic-application.xml`)
- the default value of 120 seconds

Workaround

Configure `jdbc-connection-timeout-secs` in the session descriptor.

2.33.3 Database Connections Become Unstable When a PoolLimitSQLException Occurs

Platform: All

When a `PoolLimitSQLException` occurs during a JDBC persistence session, connections to the database become unstable, and may fail with recovery or fail without recovery. This results in the loss of session data. Either an older session or null is returned.

2.33.4 Web Page Fails to Open When Accessing It Using the SSL Port

Platform: All

When accessing a Web page using the SSL port, the page fails to open and the following error is reported:

```
Secure Connection Failed

An error occurred during a connection to <hostname>.

You have received an invalid certificate. Please contact the server
administrator or email correspondent and give them the following information:

Your certificate contains the same serial number as another certificate
issued by the certificate authority. Please get a new certificate containing a
unique serial number.
```

Workaround

The following workaround can be used for Firefox.

If you have received this error and are trying to access a web page that has a self-signed certificate, perform the following steps in Firefox:

1. Go to **Tools** > **Options** >**Advanced** > **Encryption tab** > **View Certificates**.
2. On the **Servers** tab, remove the certificates.
3. On the **Authorities** tab, find the Certificate Authority (CA) for the security device that is causing the issue, and then delete it.

If you are using Internet Explorer or other web browsers, you can ignore the Warning page that appears and continue to the web page.

2.33.5 Unable to View the Output of a JSPX Page in Internet Explorer

Platform: MS Windows

When a JSPX page is deployed and is then accessed using some versions of Internet Explorer, the XHTML source is displayed instead of the page contents. This occurs in both normal and `osjp.next` modes.

Workaround

The application users should be instructed to use Firefox or Safari to access the application.

2.33.6 Unable to View the Output of SVG files in Internet Explorer 7

Platform: MS Windows

When a page using Scalar Vector Graphics is deployed and is then accessed using Internet Explorer 7 (IE7), the source is displayed instead of the page's graphic contents. This occurs in both normal and osjp.next modes.

Workaround

Application developers should avoid using SVG graphics in their applications, as it is not natively supported in IE7. If used, a warning similar to the following should be added:

```
All current browsers, with the exception of Internet Explorer, support SVG
files. Internet Explorer requires a plug-in to display SVG files. The plug-ins
are available for free, for example, the Adobe SVG Viewer at
http://www.adobe.com/svg/viewer/install/.
```

2.33.7 Changed the Expires Format For the Cookie Date in Version 0

Platform: All

WebLogic Server supports both Version 0 (Netscape) and Version 1 (RFC2109) for reading and writing a cookie expires date. WebLogic Server was previously reading and writing Version 0 cookie expires dates using the following format: *EEEE, dd-MMM-yyyy HH:mm:ss zz*. This format is incorrect and has been updated to the following: *EEE, dd-MMM-yyyy HH:mm:ss zz*.

WebLogic Server now writes using this correct formatting (*EEE, dd-MMM-yyyy HH:mm:ss zz*), but is still able to read cookie expires dates in the following formats:

- *EEE, dd-MMM-yyyy HH:mm:ss zz*
- *EEE, dd MMM yyyy HH:mm:ss zz*
- *EEEE, dd-MMM-yyyy HH:mm:ss zz*
- *EEEE, dd MMM yyyy HH:mm:ss zz*

2.34 WebLogic Server Scripting Tool (WLST) Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.34.1, "Permission Denied Error Occurs for WLST Offline Logging"
- Section 2.34.2, "Property Names Containing '.' Characters Are Not Supported by loadProperties"
- Section 2.34.3, "Invalid cachedir Created by Jython Causes WLST to Error Out"
- Section 2.34.4, "WLST returnType='a' Option Returns Child Management Objects"
- Section 2.34.5, "Domain Creation Fails on Windows Server 2012"

2.34.1 Permission Denied Error Occurs for WLST Offline Logging

When there are multiple processes, owned by different filesystem users, that are performing concurrent WLST offline operations, a `FileNotFoundException`, `Permission Denied` error may occur.

Workaround

To avoid collisions on log file names, set the following property in the environment prior to invoking `wlst.sh` *`script_name`*:

```
export WLST_PROPERTIES="-Dwlst.offline.log=./logs/filename.log"
```

Substitute a unique name for *`filename`*. You must you use a unique name for each log file to ensure that there will be no log file name collisions.

2.34.2 Property Names Containing '.' Characters Are Not Supported by loadProperties

Platform: All

The WLST `loadProperties` command does not support loading a property with a name that contains "." characters. For example, if the property `myapp.db.default` is present in the property file, WLST throws a name exception:

```
Problem invoking WLST - Traceback (innermost last):
  File "<iostream>", line 7, in ?
  File "<iostream>", line 4, in readCustomProperty
NameError: myapp
```

This is a system limitation of Python and the `loadProperties` command. WLST reads the variable names and values and sets them as variables in the Python interpreter. The Python interpreter uses "." as a delimiter to indicate module scoping for the namespace, or package naming, or both. Therefore, the properties file fails because `myapp.db.default.version=9i` is expected to be in the `myapp.db.default` package. This package does not exist.

Workaround

Use variable names that do not have periods. This will allow you to load the variables from the property file and refer to them in WLST scripts. You could use another character such as "_" or lowercase/uppercase character to delimit the namespace.

As an alternative, you can set variables from a properties files. When you use the variables in your script, during execution, the variables are replaced with the actual values from the properties file. For example:

```
myapp.py
var1=10
var2=20
import myapp
print myapp.var1
10
print myapp.var2
20
```

This will work for one level of namespaces (`myapp.var1`, `myapp.var2`). It will not work for top level variables that share the same name as the namespace (for example, `myapp=oracle` and `myapp.var1=10`). Setting the `myapp` variable will override the `myapp` namespace.

If you need multiple levels, then you can define a package namespace using directories. Create a myapp/db/default directory with a `vars.py` file as follows:

```
var1=10
var2=20
```

Then import:

```
import myapp.db.default.vars
print myapp.db.default.vars.var1
10
```

You may need to add `__init__.py` files to the subdirectories. Refer to the Python documentation for more information on packages:

https://www.python.org/doc/

2.34.3 Invalid cachedir Created by Jython Causes WLST to Error Out

Platform: All

The default `cachedir` created by Jython 2.2 is not a valid directory. If you are using Jython directly from `weblogic.jar`, this causes WLST to error out.

Workaround

There are two workarounds for this issue:

- When invoking WLST, specify the `-Dpython.cachedir=<`*`valid_directory`*`>` parameter, or
- Install Jython 2.2.1 separately instead of using the partial Jython that is included in `weblogic.jar`.

2.34.4 WLST returnType='a' Option Returns Child Management Objects

Platform: Linux

The WLST `returnType='a'` option should only return attributes from the specified directory. Instead it also returns child management objects. For example:

```
ls('Server')
drw-   AdminServer
drw-   worker01

ls('Server', returnMap='true', returnType='a')
drw-   AdminServer
drw-   worker01

ls('Server', returnMap='true',returnType='c')
drw-   AdminServer
drw-   worker01
```

The `ls` with `returnType='a'` should not list any child management objects, but `AdminServer` and `worker01` are children.

Workaround

When processing the output from `ls(returnType='a')`, check to see if the returned entry is a directory.

2.34.5 Domain Creation Fails on Windows Server 2012

Platform: Windows

Domain creation with WLST fails when running WebLogic Server 10.3.6 on Windows 2012 if “Windows Server 2012” is not in the list of supported Windows versions in the `javashell.py` file.

Workaround

Download and apply the appropriate patch from My Oracle Support at `http://support.oracle.com`. The following patches are available:

- WebLogic Server 10.3.6 – NPM3
- WebLogic Server 10.3.6.0.1 – YMQY
- WebLogic Server 10.3.6.0.8 – GXHK
- WebLogic Server 10.3.6.0.9 – IFNW

- WebLogic Server 10.3.6.0.10 – DL4N

2.35 Web Server Plug-Ins Issues and Workarounds

This section describes the following issue:

- Section 2.35.1, "MOD_WLS_OHS Does Not Fail Over"

2.35.1 MOD_WLS_OHS Does Not Fail Over

Platform: All

Currently, `mod_wl` and `mod_wl_ohs` only support container level failover and not application level failover. `mod_wl_ohs` continues to route requests to a down application as long as the managed server is up and running. In the clustered case, requests continue to go to the container where the original session started even when the application is shutdown, typically resulting in the http error 404.

2.36 Web Services and XML Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.36.1, "weblogic.wsee.jaxws.mdb.DispatchPolicy WorkManager Cannot Be Found"
- Section 2.36.2, "Multiple Resize Buffer Calls Occur"
- Section 2.36.3, "Troubleshooting Problems When Applying the WebLogic Advanced Web Services for JAX-WS Extension Template"
- Section 2.36.4, "Sparse Arrays and Partially Transmitted Arrays Are Not Supported"
- Section 2.36.5, "WSDL Compiler Does Not Generate Serializable Data Types"
- Section 2.36.6, "Use of Custom Exception on a Callback"
- Section 2.36.7, "Cannot Use JMS Transport in an Environment That Also Uses a Proxy Server"
- Section 2.36.8, "clientgen Fails When Processing a WSDL"
- Section 2.36.9, "JAX RPC Handlers in Callback Web Services Are Not Supported"
- Section 2.36.10, "Message-level Security in Callback Web Services Is Not Supported"
- Section 2.36.11, "Handling of Java Method Arguments or Return Parameters That Are JAX-RPC-style JavaBeans"
- Section 2.36.12, "IllegalArgumentException When Using a Two-Dimensional XML Object in a JWS Callback"
- Section 2.36.13, "Using SoapElement[] Results in Empty Array"
- Section 2.36.14, "FileNotFound Exception When a Web Service Invokes Another Web Service"
- Section 2.36.15, "Client Side Fails to Validate the Signature on the Server Response Message"
- Section 2.36.16, "xmlcatalog Element Entity Cannot Be a Remote File or a File in an Archive"

2.36.1 weblogic.wsee.jaxws.mdb.DispatchPolicy WorkManager Cannot Be Found

Platform: All

In some situations, warning messages are logged indicating that the weblogic.wsee.jaxws.mdb.DispatchPolicy WorkManager cannot be found, although this WorkManager is targeted to one or more of the Managed Servers in the domain.

Workaround

Use one of the following workarounds to resolve this issue.

- To prevent these warning messages, start the WebLogic Server instance with the `-Dweblogic.wsee.skip.async.response=true` flag. See *Programming Advanced Features of JAX-RPC Web Services for Oracle WebLogic Server* for more information on this flag.
- Manually target the `weblogic.wsee.jaxws.mdb.DispatchPolicy` WorkManager to the Administration Server.

2.36.2 Multiple Resize Buffer Calls Occur

Platform: All

When executing Web services client calls where Message Transmission Optimization Mechanism (MTOM) attachments are processed for send, multiple resize buffer calls occur.

Workaround

There is a patch available to resolve this issue. This patch can be applied only to WebLogic Server 10.3.4. It provides the system property `jaxws.transport.streaming`, which enables or disables streaming at the transport layer for a Web services client. Set this property to `true` for CPU-intensive applications that are running on a WebLogic Server instance that is participating in Web services interactions as a client, and is sending out large messages.

To obtain the patch, do one of the following:

- Contact My Oracle Support and request the patch for bug 9956275, or

- Download the patch from My Oracle Support and install it using Smart Update per the instructions in the following My Oracle Support document:

  `1302053.1`

  Search for Oracle patch number 9956275 or Smart Update patch 7Z5H.

2.36.3 Troubleshooting Problems When Applying the WebLogic Advanced Web Services for JAX-WS Extension Template

Platform: All

After upgrading from WebLogic Server 10.3.4 to 10.3.5, when creating or extending a domain using the WebLogic Advanced Web Services for JAX-WS Extension template (`wls_webservices_jaxws.jar`), you may encounter an exception during the execution of the final.py script. For complete details and a workaround, see "Troubleshooting Problems When Applying the WebLogic Advanced Services for JAX-WS Extension Template" in *Getting Started With JAX-WS Web Services for Oracle WebLogic Server*.

2.36.4 Sparse Arrays and Partially Transmitted Arrays Are Not Supported

Platform: All

WebLogic Server does not support Sparse Arrays and Partially Transmitted Arrays as required by the JAX-RPC 1.1 Spec.

2.36.5 WSDL Compiler Does Not Generate Serializable Data Types

Platform: All

The Web Service Description Language (WSDL) compiler does not generate serializable data types, so data cannot be passed to remote EJBs or stored in a JMS destination.

2.36.6 Use of Custom Exception on a Callback

Platform: All

WebLogic Server does not support using a custom exception on a callback that has a package that does not match the target namespace of the parent Web Service.

Workaround

Make sure that any custom exceptions that are used in callbacks are in a package that matches the target namespace of the parent Web service.

2.36.7 Cannot Use JMS Transport in an Environment That Also Uses a Proxy Server

Platform: All

You cannot use JMS transport in an environment that also uses a proxy server. This is because, in the case of JMS transport, the Web Service client always uses the t3 protocol to connect to the Web Service, and proxy servers accept only HTTP/HTTPS.

2.36.8 clientgen Fails When Processing a WSDL

Platform: All

`clientgen` fails when processing a WSDL that uses the complex type `http://www.w3.org/2001/XMLSchema{schema}` as a Web Service parameter.

2.36.9 JAX RPC Handlers in Callback Web Services Are Not Supported

Platform: All

WebLogic Server 9.2 and later does not support JAX RPC handlers in callback Web Services.

Workaround

If JAX RPC handlers were used with Web Services created with WebLogic Workshop 8.1, then such applications must be redesigned so that they do not use callback handler functionality.

2.36.10 Message-level Security in Callback Web Services Is Not Supported

Platform: All

WebLogic Server 9.2 and later does not support message-level security in callback Web Services.

Workaround

Web Services created with WebLogic Workshop 8.1 that used WS-Security must be redesigned to not use message-level security in callbacks.

2.36.11 Handling of Java Method Arguments or Return Parameters That Are JAX-RPC-style JavaBeans

Platform: All

WebLogic Server does not support handling of Java method arguments or return parameters that are JAX-RPC-style JavaBeans that contain an `XmlBean` property. For example, applications cannot have a method with a signature like this:

```
void myMethod(myJavaBean bean);
```

where `myJavaBean` class is like:

```
public class MyJavaBean {
  private String stringProperty;
  private XmlObject xmlObjectProperty;

  public MyJavaBean() {}
  String getStringProperty() {
    return stringProperty;
  }
  void   setStringProperty(String s) {
    stringProperty = s;
  }
  XmlObject getXmlObjectProperty() {
    return xmlObjectProperty;
    }
  void      getXmlObjectProperty(XmlObject x) {
    xmlObjectProperty = x;
  }
}
```

Workaround

Currently there is no known workaround for this issue.

2.36.12 IllegalArgumentException When Using a Two-Dimensional XML Object in a JWS Callback

Platform: All

Using a two dimensional XmlObject parameter (XmlObject[][]) in a JWS callback produces an IllegalArgumentException.

Workaround

Currently there is no known workaround for this issue.

2.36.13 Using SoapElement[] Results in Empty Array

Platform: All

Using SoapElement[] as a Web Service parameter with @WildcardBinding(className="javax.xml.soap.SOAPElement[]", binding=WildcardParticle.ANYTYPE) will always result in an empty array on the client.

Workaround

Do not use the @WildcardBinding annotation to change the default binding of SOAPElement[] to WildcardParticle.ANYTYPE. The SOAPElement[] default binding is set to WildcardParticle.ANY.

2.36.14 FileNotFound Exception When a Web Service Invokes Another Web Service

Platform: All

When Web Service A wants to invoke Web Service B, Web Service A should use the @ServiceClient annotation to do this. If Web Service B needs a custom policy file that is not attached to the WSDL for Web Service B, then Web Service A will fail to run. Web Service A will look for the policy file at /Web-Inf/classes/policies/*filename*.xml. Since no policy file exists at that location, WebLogic Server will throw a 'file not found' exception.

Workaround

Attach the custom policy file to Web Service B, as in this example:

```
@Policy(uri="CustomPolicy.xml",
        attachToWsdl=true)
public class B {
  ...
}
```

2.36.15 Client Side Fails to Validate the Signature on the Server Response Message

Platform: All

When the security policy has one of these Token Assertions, the client side may fail to validate the signature on the server response message.

```
  <sp:WssX509PkiPathV1Token11/>
  <sp:WssX509Pkcs7Token11/>
  <sp:WssX509PkiPathV1Token10/>
  <sp:WssX509Pkcs7Token10/>
```

In addition, when there are more than two certifications in the chain for X509 certification for <sp:WssX509Pkcs7Token11/> or <sp:WssX509Pkcs7Token10/> Token Assertion, the server side may fail to validate the signature on the incoming message.

A policy such as the following policy is not supported, unless the entire certificate chain remains on the client side.

```
<sp:AsymmetricBinding>
   <wsp:Policy>
      <sp:InitiatorToken>
         <wsp:Policy>
            <sp:X509Token
               sp:IncludeToken='. . ./IncludeToken/AlwaysToRecipient'>

            <wsp:Policy>
               <sp:WssX509Pkcs7Token11/>
            </wsp:Policy>
         </sp:X509Token>
      </wsp:Policy>
      </sp:InitiatorToken>
      <sp:RecipientToken>
      <wsp:Policy>
      <sp:X509Token sp:IncludeToken='. . ./IncludeToken/Never'>
            <wsp:Policy>
               <sp:WssX509Pkcs7Token11/>
            </wsp:Policy>
         </sp:X509Token>
      </wsp:Policy>
      </sp:RecipientToken>
   . . .
      </wsp:Policy>
   </sp:AsymmetricBinding>
```

Workaround

Use either of the following two solutions:

1. Configure the response with the `<sp:WssX509V3Token10/>` Token Assertion, instead of `WssX509PkiPathV1Token11/>`. The policy will look like this:

```
<sp:AsymmetricBinding>
   <wsp:Policy>
     <sp:InitiatorToken>
        <wsp:Policy>
        <sp:X509Token sp:IncludeToken='. . ./IncludeToken/AlwaysToRecipient'>
           <wsp:Policy>
              WssX509PkiPathV1Token11/>
           </wsp:Policy>
        </sp:X509Token>
        </wsp:Policy>
     </sp:InitiatorToken>
     <sp:RecipientToken>
        <wsp:Policy> sp:IncludeToken='. . ./IncludeToken/Never'>
        <sp:X509Token
           <wsp:Policy>
              <sp:WssX509V3Token10/>
           </wsp:Policy>
        </sp:X509Token>
        </wsp:Policy>
     </sp:RecipientToken>
 . . .
     </wsp:Policy>
```

```
</sp:AsymmetricBinding>
```

2. Configure the response with the `WssX509PkiPathV1Token11/>` token assertion, but include it in the message. The policy will look like this:

```
<sp:AsymmetricBinding>
  <wsp:Policy>
    <sp:InitiatorToken>
       <wsp:Policy>
       <sp:X509Token sp:IncludeToken='. . ./IncludeToken/AlwaysToRecipient'>
       <wsp:Policy>
          WssX509PkiPathV1Token11/>
       </wsp:Policy>
       </sp:X509Token>
    </wsp:Policy>
    </sp:InitiatorToken>
    <sp:RecipientToken>
       <wsp:Policy>
       <sp:X509Token sp:IncludeToken='. . ./IncludeToken/AlwaysToInitiator'>
          <wsp:Policy>
             WssX509PkiPathV1Token11/>
           </wsp:Policy>
       </sp:X509Token>
       </wsp:Policy>
    </sp:RecipientToken>
. . .
  </wsp:Policy>
</sp:AsymmetricBinding>
```

When there are multiple certifications in the X509 Certificate chain, `WssX509PkiPathV1Token11/>` or `<sp:WssX509PkiPathV1Token10/>` should be used, instead of `<sp:WssX509Pkcs7Token11/>` or `<sp:WssX509Pkcs7Token10/>`.

2.36.16 xmlcatalog Element Entity Cannot Be a Remote File or a File in an Archive

Platform: All

For the `xmlcatalog` element in `build.xml`, the location of an entity must be a file on the local file system. It cannot be a remote file (for example, `http:`) or a file in an archive (for example, `jar:`).

Workaround

If necessary, define the remote element as an entity in a catalog file instead.

2.36.17 Catalog File's public Element Is Not Supported When Using XML Catalogs

Platform: All

The `public` element in a catalog file is not supported when using the XML Catalogs feature. It is not supported to be consistent with JAX-WS EntityResolver implementation. WebLogic Server only supports defining the `system` element in a catalog file.

2.36.18 Local xmlcatalog Element Does Not Work Well

Platform: All

The local `xmlcatalog` element does not work well due to an Ant limitation.

Workaround

In the ant build.xml file, you have to define a local element above a clientgen(wsdlc) task when you are in the same target, or define the element out of any targets.

2.36.19 JAXRPC Client Does Not Encode the HTTP SOAPAction Header With Multi-byte Characters

Platform: All

The WebLogic Server Web Service JAXRPC client doesn't encode the HTTP SOAPAction header with multi-byte characters, but WebLogic Server only supports ASCII for HTTP headers.

Workaround

Change the SOAP action to ASCII in the WSDL.

2.36.20 External Catalog File Cannot Be Used in the xmlcatalog Element of clientgen

Platform: All

An external catalog file cannot be used in the xmlcatalog element of a clientgen task. For example, this snippet of an ant build file will not work:

```
<clientgen ...
  <xmlcatalog>
    <catalogpath>
      <pathelement location='wsdlcatalog.xml'/>
    </catalogpath>
  </xmlcatalog>
```

This is a limitation of the Ant XML Catalog.

Workaround

Resource locations can be specified either in-line or in an external catalog file(s), or both. In order to use an external catalog file, the xml-commons resolver library (resolver.jar) must be in your classpath. External catalog files may be either plain text format or XML format. If the xml-commons resolver library is not found in the classpath, external catalog files, specified in <catalogpath> paths, will be ignored and a warning will be logged. In this case, however, processing of inline entries will proceed normally.

Currently, only <dtd> and <entity> elements may be specified inline. These correspond to the OASIS catalog entry types PUBLIC and URI respectively.

2.36.21 Exceptions When Running Reliable Messaging Under Heavy Load

Platform: All

When running a Web services reliable messaging scenario under heavy load with file based storage that has the Direct-Write synchronous write policy setting, you may encounter IO exceptions similar to the following in the WebLogic Server log:

```
weblogic.store.PersistentStoreRuntimeException: [Store:280029]The
persistent store record <number> could not be found
```

or

```
Could not load conversation with id uuid:<some ID> -> Conversation read
```

```
failed:
    ...
    weblogic.wsee.jws.conversation.StoreException:
      Conversation read failed: id=uuid:<some ID>
         weblogic.store.PersistentStoreException: [Store:280052]The
         persistent store was not able to read a record.
           java.io.OptionalDataException
```

These exceptions are known to occur only when using Web Services reliable messaging. They indicate a failure to read a record from the file store and are considered 'fatal' data access errors.

The underlying issue causing these errors will be addressed in a future release.

Workaround

The following workarounds are available for this issue:

- Change the file store synchronous write policy to `Direct-Write-With-Cache`

  or

- Change the file store synchronous write policy to `Cache-Flush`.

  or

- Keep the Direct-Write synchronous write policy and add the following Java system property to your WebLogic server startup scripts:

  ```
  -Dweblogic.store.AvoidDirectIO=true
  ```

> **Note:** The `-Dweblogic.store.AvoidDirectIO` system property has been deprecated in WebLogic Server 10.3.4. Oracle recommends configuring the store synchronous write policy to `Direct-Write-With-Cache` instead.

The `Direct-Write-With-Cache` option may improve performance; it creates additional files in the operating system's temporary directory by default.

The `Cache-Flush` and `AvoidDirectIO` workarounds may lead to some performance degradation; it may be possible to reduce or eliminate the degradation by configuring a different block-size for the file store.

For important information about these settings and additional options, see "Tuning File Stores" in *Performance and Tuning for Oracle WebLogic Server*.

2.36.22 ClassNotFound Exception Occurs When Using wseeclient.jar

Platform: All

Stand-alone JAX-WS clients are not supported in this release.

Workaround

Use the client-side JAX-WS 2.1 that is integrated with the Java Standard Edition Release 6 (JDK 1.6), Update 4 and later. This requires using the JAX-WS API instead of any WebLogic Server specific APIS.

Current releases of JDK 1.6 are available for download at `http://www.oracle.com/technetwork/java/javase/downloads/index.html`. For information about writing a standalone JAX WS 2.1 client application, see the *JAX-WS*

Users Guide on the JAX-WS 2.1 Reference Implementation Web site at `https://jax-ws.java.net/`.

2.36.23 Incomplete Configuration When Adding Advanced Web Services Component to SOA Domain

Platform: All

An incomplete configuration can result when you use the Configuration Wizard to add the WebLogic Server Advanced Web Services component to a newly created SOA domain. If you create a cluster that contains only the default 'out-of-the-box' soa_server1 server definition, the resulting cluster does not include the resources needed to run WebLogic Server Web Services in that cluster.

Workaround

Use either of the following workarounds for this issue:

1. While running Configuration Wizard, create a second server in the cluster:
   a. On the **Select Optional Configuration** screen, select **Managed Servers, Clusters, and Machines**.
   b. On the **Configure Managed Servers** screen, add a managed server.
   c. On the **Assign Servers to Clusters** screen, add this server to the cluster in which the default soa_server1 server resides.
2. On the **Configuration Wizard Target Services to Servers or Clusters** screen, target Web Services resources (for example, WseeJmsServer, WseeJmsModule) to the cluster.

Either of these workarounds will cause the Configuration Wizard to apply the resources for the WebLogic Server Advanced Web Services component to the cluster.

2.36.24 WS-AT Interoperation Issues With WebSphere and WebLogic Server

Platform: All

Web Services Atomic Transactions (WS-AT) 1.1 interoperation using WebSphere as the client and either WebLogic Server or JRF as the service does not work.

WS-AT 1.1 interoperation does work when WebSphere is the service and either WebLogic Server or JRF is the client. In this case, interoperation works only if you have WebSphere 7 with Fix/Feature Pack 7.

2.37 WebLogic Tuxedo Connector Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.37.1, "View Classes are not Set on a Per Connection Basis"

2.37.1 View Classes are not Set on a Per Connection Basis

Platform: All

View classes are not set on a per connection basis.

A shared WebLogic Tuxedo Connector hash table can cause unexpected behavior in the server if two applications point to the same VIEW name with different definitions.

There should be a hash table for the view classes on the connection as well as for the Resource section.

Workaround

Ensure that all VIEW classes defined across all your WebLogic Workshop applications are consistent, meaning that you have the same VIEW name representing the same VIEW class.

2.38 Documentation Errata

This section describes documentation errata:

- Section 2.38.1, "Issues With Search Function in the Samples Viewer"
- Section 2.38.2, "Japanese Text Displays in Some Search Results Topics Avitek Medical Records"
- Section 2.38.3, "HTML Pages For Downloaded Libraries Do Not Display Properly"
- Section 2.38.4, "Evaluation Database Component Is Not Listed For silent.xml"
- Section 2.38.5, "Instructions for Reliable SOAP Messaging Code Example Are Incorrect"
- Section 2.38.6, "Documentation Incorrect for the enable-global-access-to-classes Element"

2.38.1 Issues With Search Function in the Samples Viewer

The **Search** function in the Samples viewer does not work when accessing the Examples documentation by selecting **Oracle Weblogic > Weblogic Server > Examples > Documentation** from the Windows **Start** menu.

Workaround

To search the Sample Applications and Code Examples, you must start the Examples server and navigate to `http://localhost:7001/examplesWebApp/docs/core/index.html`. Click **Instructions** and then **Search**.

2.38.2 Japanese Text Displays in Some Search Results Topics Avitek Medical Records

The samples viewer **Search** function may sometimes return topics that display the Japanese and English versions of some Avitek Medical Records topics simultaneously.

2.38.3 HTML Pages For Downloaded Libraries Do Not Display Properly

After extracting the WebLogic Server documentation library ZIP files that are available from `http://www.oracle.com/technetwork/middleware/weblogic/documentation/index.html`, the HTML pages may not display properly in some cases for the following libraries:

- E12840_01 (WebLogic Server 10.3.0 documentation library)
- E12839_01 (Weblogic Server 10.3.1 documentation library)
- E14571_01 (WebLogic Server 10.3.3 documentation library)

Workarounds

For library E12840-01, after extracting the E12840_01.zip library file, if the HTML pages are not formatting correctly, perform the following steps:

1. Go to the directory in which you extracted the zip file.
2. Locate the `/global_resources` directory in the directory structure.
3. Copy the `/global_resources` directory to the root directory of the same drive.

For libraries E12839-01 and E14571-01, this issue occurs only on Windows operating systems. If the HTML pages of the extracted library are not formatting correctly, try extracting the ZIP file using another extraction option in your unzip utility. For example, if you are using 7-Zip to extract the files, select the **Full pathnames** option. Note that you cannot use the Windows decompression utility to extract the library ZIP file.

2.38.4 Evaluation Database Component Is Not Listed For silent.xml

In the *WebLogic Server Installation Guide*s for WebLogic Server 10.3.3 and 10.3.4, the Evaluation Database is not listed as an installable component in Table 5-1 of Chapter 5, "Running the Installation Program in Silent Mode.:" The following entry should be included in the Component Paths row:

```
WebLogic Server/Evaluation Database
```

The Evaluation Database component is automatically installed if the Server Examples component is included in `silent.xml`. Therefore, it does not have to be explicitly included in `silent.xml`. If, however, you do not install the Server Examples, but you want to install the Evaluation Database, you must include `WebLogic Server/Evaluation Database` in `silent.xml`.

2.38.5 Instructions for Reliable SOAP Messaging Code Example Are Incorrect

The instructions for the "Configuring Secure and Reliable SOAP Messaging for JAXWS Web Services" example are a copy of the instructions for the "Using Make Connection and Reliable Messaging for JAX-WS Web Service" example.

The correct instructions for the "Configuring Secure and Reliable SOAP Messaging for JAXWS Web Services" example are provided here.

2.38.5.1 About the Example

This example shows how to configure secure, reliable messaging for JAX-WS Web services. The example includes the following WebLogic Web services:

- Web service whose operations can be invoked using reliable and secure SOAP messaging (*destination endpoint*).
- Client Web service that invokes an operation of the first Web service in a reliable and secure way (*source endpoint*).

Overview of Secure and Reliable SOAP Messaging

Web service reliable messaging is a framework that enables an application running on one application server to reliably invoke a Web service running on another application server, assuming that both servers implement the WS-RelicableMessaging specification. Reliable is defined as the ability to guarantee message delivery between the two endpoints (Web service and client) in the presence of software component, system, or network failures.

WebLogic Web services conform to the WS-ReliableMessaging 1.2 specification (February 2009) and support version 1.1. This specification describes how two endpoints (Web service and client) on different application servers can communicate reliably. In particular, the specification describes an interoperable protocol in which a message sent from a *source endpoint* (or client Web service) to a *destination endpoint* (or Web service whose operations can be invoked reliably) is guaranteed either to be delivered, according to one or more delivery assurances, or to raise an error.

WebLogic Web services use WS-Policy files to enable a destination Web service to describe and advertise its reliable SOAP messaging capabilities and requirements. WS-Policy files are XML files that describe features such as the version of the WS-ReliableMessaging specification that is supported, the source Web service retransmission interval, the destination Web service acknowledgment interval, and so on.

Overview of the Example

This example uses JWS annotations to specify the shape and behavior of the Web services. It describes additional JWS annotations to enable reliable and secure SOAP messaging in the destination Web service and to reliably invoke an operation from the source Web service in a secure way.

The destination `ReliableEchoService` Web service has two operations that can be invoked reliably and in a secure way: `echo` and `echoOneway`. The JWS file that implements this Web service uses the `@Policies` and `@Policy` JWS annotations to specify the WS-Policy file, which contains the reliable and secure SOAP messaging assertions.

The source `ClientService` Web service has one operation for invoking the echo operations of the `ReliableEchoService` Web service reliably and in a secure way within one conversation: `runTestEchoWithRes`. The JWS file that implements the ClientService Web service uses the `@WebServiceRef` JWS annotation to specify the service name of the reliable Web service being invoked.

To generate the Web services, use the `jwsc` WebLogic Web service Ant task, as shown in the build.xml file. The `jwsc` target generates the reliable and secure Web service and the `jwsc-client-app` target creates the source Web service that invoke the echo operations of the `ReliableEchoService` Web service. The `jwsc` Ant task compiles the JWS files, and generates the additional files needed to implement a standard Java EE Enterprise Web service, including the Web service deployment descriptors, the WSDL file, data binding components, and so on. The Ant task automatically generates all the components into an Enterprise Application directory structure that you can then deploy to WebLogic Server. This example uses the wldeploy WebLogic Ant task to deploy the Web service.

The `jwsc-client-app` target also shows how you must first execute the `clientgen` Ant task to generate the JAX-WS stubs for the destination `ReliableEchoService` Web service, compile the generated Java source files, and then use the `classpath` attribute of `jwsc` to specify the directory that contains these classes so that the `ClientServiceImpl.java` class can find them.

The `WsrmJaxwsExampleRequest.java` class is a standalone Java application that invokes the `echo` operation of the source Web service. The client target of the `build.xml` file shows how to run `clientgen`, and compile all the generated Java files and the `WsrmJaxwsExampleRequest.java` application.

2.38.5.2 Files Used in This Example

Directory Location: *MW_HOME*/wlserver_10.3/samples/server/examples/src/examples/webservices/wsrm_jaxws/wsrm_jaxws_security

MW_HOME represents the Oracle Fusion Middleware home directory.

File	Description
ClientServiceImpl.java	JWS file that implements the source Web service that reliably invokes the echo operation of the `ReliableEchoService` Web service in a secure way.
ReliableEchoServiceImpl.java	JWS file that implements the reliable destination Web service. This JWS file uses the `@Policies` and `@Policy` annotation to specify a WS-Policy file that contains reliable and secure SOAP messaging assertions.
client/WsrmJaxwsExampleRequest.java	Standalone Java client application that invokes the source WebLogic Web service, that in turn invokes an operation of the `ReliableEchoervice` Web service in a reliable and secure way.
ws_rm_configuration.py	WLST script that configures the components required for reliable SOAP messaging. Execute this script for the WebLogic Server instance that hosts the reliable destination Web service. The out-of-the-box Examples server has already been configured with the resources required for the source Web service that invokes an operation reliably.
configWss.py	WLST script that configures the components required for secure SOAP messaging. Execute this script for the WebLogic Server instance that hosts the source Web service. Remember to restart the source WebLogic Server after executing this script.
configWss_Service.py	WLST script that configures the components required for secure SOAP messaging. Execute this script for the WebLogic Server instance that hosts the destination Web service. Remember to restart the destination WebLogic Server after executing this script.
certs/serverKeyStore.jks	Server-side key store used to create the server-side `BinarySecurityToken` credential provider.
certs/clientKeyStore.jks	Client-side key store used to create the client-side `BinarySecurityToken` credential provider.
jaxws-binding.xml	XML file that describes the package name of the generated code and indicate the client side code needs to contain asynchronous invocation interface.
build.xml	Ant build file that contains targets for building and running the example.

2.38.5.3 Prepare the Example

This section describes how to prepare the example.

Prerequisites

Before working with this example:

1. Install Oracle WebLogic Server, including the examples.
2. Start the Examples Server.
3. Set up your environment.

Configure the Destination WebLogic Server Instance (Optional)

The default configuration for this example deploys both the source and destination Web services to the Examples server. You can use this default configuration to see how the example works, but it does not reflect a real life example of using reliable and secure SOAP messaging in which the source Web service is deployed to a WebLogic Server that is *different* from the one that hosts the destination Web service. This section describes how to set up the real life example.

The example includes WebLogic Server Scripting Language (WLST) scripts that are used to configure:

- Store-and-forward (SAF) service agent
- File store
- JMS server
- JMS module
- JMS subdeployment
- JMS queues
- Logical store
- Credential provider for Security Context Token
- Credential provider for Derived Key
- Credential provider for x.509
- KeyStores for Confidentiality and Integrity
- PKI CreditMapper

Follow these steps if you want to deploy the secure and reliable destination Web service to a different WebLogic Server instance:

1. If the managed WebLogic Server to which you want to deploy the reliable JAX-WS Web service does not exist, create it.
2. Change to the *SAMPLES_HOME*\server\examples\src\examples\webservices\wsrm_jaxws\wsrm_jaxws_security directory, where *SAMPLES_HOME* refers to the main WebLogic Server examples directory, such as c:\Oracle\Middleware\wlserver_10.3\samples.
3. Edit the build.xml file and update the following property definitions to ensure that the reliable JAX-WS Web service is deployed to the destination WebLogic Server:

   ```
   <property name="wls.service.server" value="destinationServerName" />
   <property name="wls.service.hostname" value="destinationHost" />
   <property name="wls.service.port" value="destinationPort" />
   <property name="wls.service.username" value="destinationUser" />
   <property name="wls.service.password" value="destinationPassword" />
   ```

   Substitute the italicized terms in the preceding properties with the actual values for your destination WebLogic Server. The default out-of-the-box build.xml sets these properties to the Examples server.

Build and Deploy the Example

To build and deploy the example:

1. Change to the *SAMPLES_HOME*\server\examples\src\examples\webservices\wsrm_jaxws\wsrm_jaxws_security directory, where *SAMPLES_HOME* refers to the main WebLogic Server examples directory, such as c:\Oracle\Middleware\wlserver_10.3\samples.
2. Run the WLST script that configures the destination WebLogic Server by executing the config.ws.reliable.service target of the build.xml file from the shell where you set your environment:

   ```
   prompt> ant config.ws.reliable.service
   ```

3. Execute the following command to configure JAX-WS Web service Security from the shell where you set your environment:

   ```
   prompt> ant config.wss
   ```

4. If you have configured a different destination WebLogic Server (that is, the destination server is not the Examples server), copy the certs\serverKeyStore.jks file to the domain directory of your destination server.
5. Restart both your client and destination WebLogic Server to activate the MBean changes.
6. Execute the following command from the shell where you set your environment:

   ```
   prompt> ant build
   ```

   This command compiles and stages the example. Specifically, it compiles both the source and destination Web services. It also compiles the standalone `WsrmJaxwsExampleRequest` application that invokes the source Web service, which in turn invokes the reliable destination Web service.
7. Execute the following command from the shell where you set your environment:

   ```
   prompt> ant deploy
   ```

   This command deploys, by default, both the source and destination Web services to the wl_server domain of your WebLogic Server installation. If you have configured a different destination WebLogic Server and updated the build.xml file accordingly, then the reliable JAX-WS Web service is deployed to the configured destination server.

2.38.5.4 Run the Example

To run the example, follow these steps:

1. Complete the steps in the Prepare the Example section.
2. In your development shell, run the `WsrmJaxwsExampleRequest` Java application using the following command from the main example directory (*SAMPLES_HOME*\server\examples\src\examples\webservices\wsrm_jaxws\wsrm_jaxws_security):

   ```
   prompt> ant run
   ```

   This command runs the standalone `WsrmJaxwsExampleRequest` application that invokes the source Web service, which in turn invokes the reliable destination JAX-WS Web service.

3. To test the reliability of the Web service, stop the destination WebLogic Server, and then rerun the `WsrmJaxwsExampleRequest` application. When you restart the destination WebLogic Server and the reliable Web service is deployed, you should see that the operation is also automatically invoked.

Check the Output

If your example runs successfully, the following messages display in the command shell from which you ran the `WsrmJaxwsExampleRequest` application:

```
Trying to override old definition of task clientgen

run:
     [java]
     [java]
     [java] ############################################
     [java]     In testEcho_AsyncOnServerClient_ServiceBuffered...
     [java]     On-Server / Async / Buffered case
     [java]     2011/06/160 03:30:29.938
     [java] ############################################
     [java]
     [java]
     [java] Client addr:http://localhost:9001/wsrm_jaxws_sc_example_client/Clien
tService
     [java] ---[HTTP request - http://localhost:9001/wsrm_jaxws_sc_example_clien
t/ClientService]---
     [java] Content-type: text/xml;charset=utf-8
     [java] Soapaction: ""
     [java] Accept: text/xml, multipart/related, text/html, image/gif, image/jpe
g, *; q=.2, */*; q=.2
     [java] <?xml version='1.0' encoding='UTF-8'?><S:Envelope xmlns:S="http://sc
hemas.xmlsoap.org/soap/envelope/"><S:Body><ns2:runTestEchoWithRes xmlns:ns2="htt
p://example.wsrm_jaxws/"><arg0>Foo bar</arg0><arg1>localhost</arg1>
<arg2>8001</arg2><arg3>C:\Oracle\Middleware\wlserver_10.3\samples\server\
examples\src\examples\webservices\wsrm_jaxws_security/certs</arg3>
</ns2:runTestEchoWithRes></S:Body></S:Envelope>--------------------
     [java]
     [java] ---[HTTP response - http://localhost:9001/wsrm_jaxws_sc_example_clie
nt/ClientService - 200]---
     [java] Transfer-encoding: chunked
     [java] null: HTTP/1.1 200 OK
     [java] Content-type: text/xml;charset=utf-8
     [java] X-powered-by: Servlet/2.5 JSP/2.1
     [java] Date: Thu, 09 Jun 2011 07:30:33 GMT
     [java] <?xml version='1.0' encoding='UTF-8'?><S:Envelope xmlns:S="http://sc
hemas.xmlsoap.org/soap/envelope/"><S:Body><ns2:runTestEchoWithResResponse xmlns:
ns2="http://example.wsrm_jaxws/"><return>[2011/06/160 03:30:33.953] ## Making Ec
ho Requests (ASYNC/BUFFERED) ##
     [java] [2011/06/160 03:30:42.703] *** On first good invoke ***
     [java] [2011/06/160 03:30:42.703] echo returned: Foo bar expected: Foo bar
     [java] [2011/06/160 03:30:42.922] echo returned: foo bar 2 expected: foo ba
r 2
     [java] [2011/06/160 03:30:43.031] echo returned: foo bar 3 expected: foo ba
r 3
     [java] [2011/06/160 03:30:43.031] ## Done Making Echo Requests (ASYNC/BUFFE
RED) ##
     [java] </return></ns2:runTestEchoWithResResponse></S:Body>
</S:Envelope>--------------------
     [java]
     [java] [2011/06/160 03:30:33.953] ## Making Echo Requests (ASYNC/BUFFERED)
##
```

```
     [java] [2011/06/160 03:30:42.703] *** On first good invoke ***
     [java] [2011/06/160 03:30:42.703] echo returned: Foo bar expected: Foo bar
     [java] [2011/06/160 03:30:42.922] echo returned: foo bar 2 expected: foo ba
r 2
     [java] [2011/06/160 03:30:43.031] echo returned: foo bar 3 expected: foo ba
r 3
     [java] [2011/06/160 03:30:43.031] ## Done Making Echo Requests (ASYNC/BUFFE
RED) ##
     [java]

BUILD SUCCESSFUL
Total time: 2 minutes 33 seconds
```

The following messages display in the command window from which you started as the client WebLogic Server (that hosts the reliable source Web service):

```
Service addr:http://localhost:7001/wsrm_jaxws_sc_example/ReliableEchoService
    [2011/06/180 01:33:40.906] ## Making Echo Requests (ASYNC/BUFFERED) ##

    [2011/06/180 01:33:40.906] In invokeEchoAsync, invoking echo with request: Foo
bar

    [2011/06/180 01:33:40.906] In invokeEchoAsync, waiting for response to
request: Foo bar ...

    SignInfo mismatch  Algo mismatch http://www.w3.org/2000/09/xmldsig#rsa-sha1
VS.
    http://www.w3.org/2000/09/xmldsig#hmac-sha1 Refs: Msg size =1#Signature_
prfr5thF
    y2vRPbpC, Policy size =3 #unt_w7HSTtcGcebXFWEr, #Timestamp_XIXttwj9Yq2XO7Tj,
#Bo
    dy_81D2x3V7iTNyy1I5,
    STR type mismatch Actual
KeyInfo:{http://docs.oasis-open.org/wss/2004/01/oasis-2

00401-wss-wssecurity-secext-1.0.xsd}KeyIdentifier|http://docs.oasis-open.org/wss
    /oasis-wss-soap-message-security-1.1#ThumbprintSHA1,  StrTypes size=1
:{http://d

ocs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd}Refere
    nce||http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk,
    Security Token mismatch, token type
=http://docs.oasis-open.org/ws-sx/ws-securec
    onversation/200512/dk and actual
ishttp://docs.oasis-open.org/wss/2004/01/oasis-
    200401-wss-x509-token-profile-1.0#X509v3
    <WSEE:15>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
    testing..................


    [2011/06/180 01:33:41.718] In ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fb8)

    [2011/06/180 01:33:41.718] Done with ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fb8): Foo bar

    [2011/06/180 01:33:41.718] *** On first good invoke ***

    [2011/06/180 01:33:41.734] echo returned: Foo bar expected: Foo bar
    [2011/06/180 01:33:41.734] In invokeEchoAsync, invoking echo with request: foo
bar 2

    [2011/06/180 01:33:41.750] In invokeEchoAsync, waiting for response to
request: foo bar 2 ...

    <WSEE:15>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
    testing..................


    [2011/06/180 01:33:41.984] In ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fae)

    [2011/06/180 01:33:41.984] Done with ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fae): foo bar 2

    [2011/06/180 01:33:41.984] echo returned: foo bar 2 expected: foo bar 2

    [2011/06/180 01:33:42.000] In invokeEchoAsync, invoking echo with request: foo
bar 3

    [2011/06/180 01:33:42.015] In invokeEchoAsync, waiting for response to
request: foo bar 3 ...

    <WSEE:31>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
    testing..................

    [2011/06/180 01:33:42.187] In ClientServiceImpl.onEchoResponse(request:
    examplesServer:4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fab)

    [2011/06/180 01:33:42.328] Done with ClientServiceImpl.onEchoResponse(request:
    examplesServer:4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fab): foo bar 3

    [2011/06/180 01:33:42.328] echo returned: foo bar 3 expected: foo bar 3

    [2011/06/180 01:33:42.328] ## Done Making Echo Requests (ASYNC/BUFFERED) ##

    <WSEE:46>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
```

The following messages display in the command window from which you started the destination WebLogic Server (that hosts the reliable destination Web service):

```
%% Echoing: Foo bar %%
%% Echoing: foo bar 2 %%
%% Echoing: foo bar 3 %%
```

If you deploy both the source and destination Web services to the same Server, the following messages display in the command window from which you started your client and destination WebLogic Server:

```
    Service addr:http://localhost:7001/wsrm_jaxws_sc_example/ReliableEchoService
    [2011/06/180 01:33:40.906] ## Making Echo Requests (ASYNC/BUFFERED) ##

    [2011/06/180 01:33:40.906] In invokeEchoAsync, invoking echo with request: Foo
bar

    [2011/06/180 01:33:40.906] In invokeEchoAsync, waiting for response to
request: Foo bar ...
```

```
    SignInfo mismatch  Algo mismatch http://www.w3.org/2000/09/xmldsig#rsa-sha1
VS.
    http://www.w3.org/2000/09/xmldsig#hmac-sha1 Refs: Msg size =1#Signature_
prfr5thF
    y2vRPbpC, Policy size =3 #unt_w7HSTtcGcebXFWEr, #Timestamp_XIXttwj9Yq2XO7Tj,
#Bo
    dy_81D2x3V7iTNyy1I5,
    STR type mismatch Actual
KeyInfo:{http://docs.oasis-open.org/wss/2004/01/oasis-2

00401-wss-wssecurity-secext-1.0.xsd}KeyIdentifier|http://docs.oasis-open.org/wss
    /oasis-wss-soap-message-security-1.1#ThumbprintSHA1,  StrTypes size=1
:{http://d

ocs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd}Refere
    nce||http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk,
    Security Token mismatch, token type
=http://docs.oasis-open.org/ws-sx/ws-securec
    onversation/200512/dk and actual
ishttp://docs.oasis-open.org/wss/2004/01/oasis-
    200401-wss-x509-token-profile-1.0#X509v3
    %% Echoing: Foo bar %%
    <WSEE:15>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
    testing..................


    [2011/06/180 01:33:41.718] In ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fb8)

    [2011/06/180 01:33:41.718] Done with ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fb8): Foo bar

    [2011/06/180 01:33:41.718] *** On first good invoke ***

    [2011/06/180 01:33:41.734] echo returned: Foo bar expected: Foo bar

    [2011/06/180 01:33:41.734] In invokeEchoAsync, invoking echo with request: foo
bar 2

    [2011/06/180 01:33:41.750] In invokeEchoAsync, waiting for response to
request: foo bar 2 ...

    %% Echoing: foo bar 2 %%
    <WSEE:15>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
    testing..................


    [2011/06/180 01:33:41.984] In ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fae)

    [2011/06/180 01:33:41.984] Done with ClientServiceImpl.onEchoResponse(request:
    examplesServer: 4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fae): foo bar 2

    [2011/06/180 01:33:41.984] echo returned: foo bar 2 expected: foo bar 2

    [2011/06/180 01:33:42.000] In invokeEchoAsync, invoking echo with request: foo
bar 3
```

```
    [2011/06/180 01:33:42.015] In invokeEchoAsync, waiting for response to
request: foo bar 3 ...

    %% Echoing: foo bar 3 %%
    <WSEE:31>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
    testing..................

    [2011/06/180 01:33:42.187] In ClientServiceImpl.onEchoResponse(request:
    examplesServer:4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fab)

    [2011/06/180 01:33:42.328] Done with ClientServiceImpl.onEchoResponse(request:
    examplesServer:4b1c0f3e575dfa8c:7291c50f:130d9cbaace:-7fab): foo bar 3

    [2011/06/180 01:33:42.328] echo returned: foo bar 3 expected: foo bar 3

    [2011/06/180 01:33:42.328] ## Done Making Echo Requests (ASYNC/BUFFERED) ##

    <WSEE:46>There is no information on the incoming SOAP message.
    <SmartPolicySelect or.getSmartPolicyBlueprint:501>
```

2.38.6 Documentation Incorrect for the enable-global-access-to-classes Element

Prior to 10.3.6, the documentation for the `enable-global-access-to-classes` subelement of the `weblogic-connector` element incorrectly states the following: When set to `false` (default), the resource adapter allows global access to its classes.

The documentation has been updated as of 10.3.6 to clarify that the resource adapter allows global access to its classes when `enable-global-access-to-classes` is set to `true`. To see the updated documentation, see "weblogic-connector" in *Programming Resource Adapters for Oracle WebLogic Server*.