Oracle® Fusion Middleware

Release Notes for Oracle WebLogic Server

12*c* (12.1.2)

E28073-10

April 2015

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

ORACLE®

Oracle Fusion Middleware Release Notes for Oracle WebLogic Server, 12*c* (12.1.2)

E28073-10

Contents

Preface

This document describes all known issues with Oracle WebLogic Server 12*c* (12.1.2).

Audience

This document is intended for all users of Oracle WebLogic Server 12*c* (12.1.2).

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.
`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 12*c* (12.1.2). It includes the following topics:

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 12*c* (12.1.2). 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 *Installing and Configuring Oracle WebLogic Server and Coherence*.

1.4 Certification Information

For certification information, see the Oracle Fusion Middleware Supported System Configurations page on Oracle Technology Network.

1.5 Licensing Information

Licensing information for Oracle WebLogic Server is available at:

http://shop.oracle.com

2

Oracle WebLogic Server

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

2.1 General Issues and Workarounds

This section describes the following issues and workarounds:

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 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`.

2.1.3 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.

2.1.4 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.

2.1.5 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
```

2.1.6 Installation Requirements if Using Coherence With Maven

Coherence users who do not have a dependency on WebLogic Server, and who want to use Maven, should use the standalone Coherence installer.

Coherence users who do have a dependency on WebLogic Server who want to use Maven must choose the "WebLogic Server" or "Complete with Examples" installation options. Do not choose the "Coherence Installation" option.

2.2 Administration Console Issues and Workarounds

This section describes the following issues and workarounds:

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 Application Testing Links Fail to Resolve in Administration Console

Platform: All

In some configurations, the Application Testing pages included in the WebLogic Server Administration Console use IPv6 addresses in the testing links. These addresses are valid for WebLogic server instances, but in some mixed IPv4 and IPv6 environments, these addresses cannot be used from the browser to interact with applications and the testing links cannot be resolved.

Workaround

This scenario typically happens when an administrator does not specify the listen address for a server in the configuration and the server is running on a dual stack (IPv6/IPv4) machine where Java and the operating system are configured to use IPv6 in preference to IPv4. In these mixed environments where the IPv4 stack cannot communicate with IPv6, Oracle recommends starting all server instances with the following command so that all servers are downgraded to use IPv4 only:

```
-Djava.net.preferIPv4Stack=true
```

2.2.8 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.

2.2.9 Error When Configuring Security Role For Newly Created Coherence Cluster Service or Cache

Platform: All

An unexpected error condition is noted in the WebLogic Server Administration Console when configuring a security role for a newly created Coherence cluster service or cache. It is a common pattern in the WebLogic Server Administration Console that

newly created artifacts must be saved and activated before it is possible to access them to configure security roles and policies on those artifacts. Many console pages check this and display a message indicating that "This page is not available because the necessary security providers have not been configured, or those configuration changes are pending and not yet activated. Please activate the changes and (if necessary) restart the Administration Server to make this page available." This check is not present in the Coherence security pages.

Workaround

After creating a new Coherence cluster, activate the configuration changes and restart any servers as indicated in the restarts changelist. This ensures that the Coherence cluster resources are available for role and policy configuration.

2.2.10 Online Help States "Start Node Manager Using a Shortcut on the Start Menu"

Platform: MS Windows

The option to start Node Manager on Windows machines using a shortcut on the Start menu has been removed in WebLogic Server 12.1.2.

Workaround

Ignore this text: "On Windows, you can start Node Manager using a shortcut on the Start menu."

Use other methods to start Node Manager. See "Starting and Stopping Node Manager" in *Administering Node Manager for Oracle WebLogic Server*.

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 issues and workarounds:

- Section 2.4.1, "Threads Are Blocked on Cluster Messaging in Unicast Mode"
- Section 2.4.2, "Exceptions Occur When Cloning Managed Servers to Expand a Large Non-Dynamic Cluster"

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.

2.4.2 Exceptions Occur When Cloning Managed Servers to Expand a Large Non-Dynamic Cluster

Platform: All

When attempting to clone Managed Servers to expand a large non-dynamic cluster, the following exceptions can occur: `java.nio.channels.CancelledKeyException` and

`java.lang.RuntimeException:MaxThreads` constraint. Additionally, the Administration Server can slow or become unresponsive.

Workaround

Choose one of the following workarounds:

- Use multicast messaging mode.
- Use server templates to create a dynamic cluster and expand the cluster. For more information, see "Creating Dynamic Clusters" in *Administering Clusters for Oracle WebLogic Server*.
- Instead of cloning Managed Servers and adding the cloned servers to a running cluster configuration, first determine the number of Managed Servers needed by the cluster, add those Managed Servers to the cluster configuration, and then start them as needed.

2.5 Configuration Issues and Workarounds

This section describes the following issues and workarounds:

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 Middleware 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 application 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%\common\bin\wlst.cmd
```

2.5.3 Configuration Tools Can Fail If WebLogic Installation Path Contains Spaces

Platform: MS 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: All

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.

2.5.8 Administration Server Memory Consumption and JMX Notifications

Platform: All

The Domain Runtime MBean Server is a federated MBean server with connections to all Managed Server Runtime MBean Servers in the domain. The federation architecture performs well with queries. However, when JMX notifications are added to MBeans, the Domain Runtime MBean Server can consume large amounts of memory.

When JMX notifications are used, two cases exist that cause the Administration Server to keep copies of all JMX object names registered in all Runtime MBean Servers running in all Managed Servers in the domain:

- At the WebLogic Server level, to simulate the unregister MBean notifications when a Managed Server shuts down.
- At the JDK JMX client notification layer.

The likelihood of encountering this issue increases when both of the following conditions exist:

- EM Fusion Middleware Control is being used to manage large domains, as it adds notification listeners to the Domain Runtime MBean Server.
- Fusion Middleware products that significantly increase the number of JMX runtime MBeans are included in the domain. This would include any product with MBeans that are registered in WebLogic Server Runtime MBean Server instances running in the domain; that is, in the Administration Server as well as all Managed Servers. (These products include Coherence, SOA Suite, OSB, and so on.)

Workaround

Disable the `managed-server-notifications-enabled` attribute. This configuration attribute disables the ability to define notifications on MBeans that are contained in the Managed Servers Runtime MBean Servers (these MBeans contain a `Location=key` in the ObjectName).

If Managed Server notifications are disabled, then the two sets of ObjectNames for MBeans contained in the WebLogic Server and JDK components will not be kept. Notifications listeners can still be defined on the MBeanServerDelegate and on MBeans contained in the local Domain Runtime MBean Server. However, notifications listeners cannot be added to the non-local MBeans.

The `managed-server-notifications-enabled` attribute can be set using WLST as follows:

```
edit()
startEdit()
cd("JMX/domain-name")
cmo.setManagedServerNotificationsEnabled(false)
```

```
activate()
```

2.5.9 Do Not Select SIP Servlet Template in the Configuration Wizard

Platform: All

The SIP Servlet domain creation template is listed as an available component in the WebLogic Server Configuration Wizard. Although visible, do not select the SIP Servlet template when configuring your WebLogic Server domain.

2.5.10 Issue Rolling Back Changes For `editCustom()` MBeans

Platform: All

The `editCustom()` tree contains MBeans for upper stack and system component products. If you make changes to these MBeans, the changes are persisted immediately to the pending directory. This is different from the WebLogic Server MBeans in the `edit()` tree, which require an explicit save.

If you use `stopEdit()`, `cancelEdit()` or exit WLST with an open edit session, then the unsaved changes to the WebLogic Server MBeans will be rolled back. However, the changes to the `editCustom()` tree will not be rolled back since they have been persisted.

Workaround

Use the `undo('y')` command to rollback the unactivated changes to the `editCustom()` MBeans.

2.5.11 WebLogic Server Shell Scripts Return "is not an identifier" Error in Solaris 5.10

Platform: Solaris 5.10 - X64 and SPARC

In Solaris X64 and Solaris SPARC platforms, an error occurs if the `export` keyword is in the same line as the equals sign (=) for the `/bin/sh` shell.

Workaround

Replace the line that contains the `export` keyword and equals sign with two separate lines. For example:

```
> ./Middleware/oracle_common/common/bin/getproperty.sh
> 26:export ${ENV_VAR}=`grep ${PROPERTY_NAME} ${PROPERTIES_FILE} | cut -d '='  -f
2`
>
> ./Middleware/wlserver/common/bin/wlsifconfig.sh
> 391: export INET6="inet6"
```

2.5.12 Issue Starting Multiple Development Servers On Same Host

Platform: All

Two development servers cannot be started on the same host if they are sharing a single Derby instance.

Workaround

Configure each development server to use its own unique instance of Derby instead of sharing a single Derby instance. For more information see "Running Each Domain with a Unique Derby Instance" in *Developer's Guide for Oracle Service Bus*.

2.5.13 Coherence Cache Override Not Working

Platform: All

If the WebLogic Server Configuration Wizard (`config.sh`) is used to create a domain and the WebLogic Coherence Cluster Extension template is specified, then a Coherence cluster will be defined. The Coherence cluster will be associated with any Managed Server or WebLogic Server cluster that is also created by the Configuration Wizard. If no Managed Server or WebLogic Server cluster is created, then the Coherence cluster will be associated with the Administration Server. This association between the Coherence cluster and the servers is not completely defined using the WebLogic Server configuration tool, which results in the Coherence cache configuration override file not being detected by the Coherence cluster. Please note that this issue only occurs if you are using the cache override feature.

Workaround

Use the following workaround:

1. Start the domain created with the Configuration Wizard and connect using the WebLogic Server Administration Console.
2. In the left pane of the Administration Console, expand **Environment** and select **Coherence Clusters**.
3. Select your Coherence cluster. The Coherence cluster settings page is displayed.
4. Select the **Members** tab, which displays all of the members of the Coherence cluster.
5. Deselect the servers and clusters that are members of the Coherence cluster and click **Save**.
6. Reselect the servers and clusters that are the desired members for the Coherence cluster and click **Save**.

This will perform a complete association between the Coherence cluster and the targeted servers, which is required to detect and utilize the specified Coherence cluster cache configuration override file.

2.5.14 1st arg can't be coerced to String Error Occurs in WLST

Platform: All

When executing a WLST script or WLST command, a `TypeError: state(): 1st arg can't be coerced to String error occurs`. This error occurs because a WLST class name `server` was used as a variable name in a WLST command. For example, in the following command, the value for the variable `server` will be replaced with its class name during classloading and will therefore cause this error:

```
state(server,'Server')
```

Workaround

Use one of the following workarounds:

- Include the `-Dpython.cachedir.skip=true` parameter when starting WLST.
- Change the reserved string name to another string. For example, you can change the string name `server` to `srvr` to resolve the issue.

2.6 Connector (Resource Adapter) Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.6.1, "java.IO.Serializable Error Occurs When Deploying a JCA Connector"

2.6.1 java.IO.Serializable Error Occurs When Deploying a JCA Connector

When deploying a WebLogic Server resource adapter, the following error occurs:

```
The activation_spec, which is defined as resource from implementing_class, must
implement java.io.Serializable but does not.
```

Workaround

Update the activation spec implementing class so that it is Serializable.

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, "java.io.Exception Occurs When Creating Multicast Socket"
- Section 2.8.2, "NodeManager, Administration Server, or Managed Servers Fails"
- Section 2.8.3, "Threads Become Stuck While Waiting to Get a Connection"
- Section 2.8.4, "Using IPv6-Formatted Addresses"
- Section 2.8.5, "Server Cannot Be Started After a Whole Server Migration"
- Section 2.8.6, "Object State is not Retained After Renaming Field"
- Section 2.8.7, "Servers Configured to Listen on a Host Name Are Listening on a Different Host Name After Startup"
- Section 2.8.8, "Administration Server or Node Manager Cannot Track the Status of a Managed Server"
- Section 2.8.9, "Multicast Traffic Observed to be Unreliable During or After a Network Partition"
- Section 2.8.10, "No Java DB Leasing Script or Support"
- Section 2.8.11, "Initial Connection May Remain Open When Using t3 Protocol with External Load Balancers"

2.8.1 java.io.Exception Occurs When Creating Multicast Socket

Platform: zLinux

A `java.io.IOException` may occur when creating a multicast socket with an IPv4-format address for a WebLogic Server cluster if the JDK or operating system is configured for IPv6 format.

Workaround

Include the `-Djava.net.preferIPv4Stack=true` parameter in the server startup command.

2.8.2 NodeManager, Administration Server, or Managed Servers Fails

Platform: AIX, Solaris X64, SPARC

When the Operating System's `ulimit` value for the number of open file descriptors is set to `unlimited`, the Node Manager, Administration Server, or Managed Servers in the domain may fail to start or may stop running.

Workaround

For the user account that is being used to start WebLogic Server, set the Operating System's `ulimit` value to something other than unlimited. For example:

```
ulimit -n 1024
```

2.8.3 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.

For more information about timeout properties, see "Setting Client Timeouts" in *Developing RMI Applications for Oracle WebLogic Server*.

2.8.4 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
```

2.8.5 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.

2.8.6 Object State is not Retained After Renaming Field

Platform: All

When FastSwap is enabled in a J2EE 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.

2.8.7 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.

2.8.8 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.

2.8.9 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.

2.8.10 No Java DB Leasing Script or Support

Platform: All

WebLogic Server does not support Java DB for migration. There is no leasing script available in the *WL_HOME*`/server/db` directory for Java DB.

There is no workaround for this issue.

2.8.11 Initial Connection May Remain Open When Using t3 Protocol with External Load Balancers

Platform: All

When using the `t3` protocol with external load balancers, the initial connection may remain associated with the IP address of the load balancer used to do the bootstrap into the cluster. Therefore, a small percentage of requests may pass through the load balancer after the initial connection. This behavior, if present, is a side-effect of the implementation that should not be relied upon since Oracle reserves the right to change this behavior at any point in time.

2.9 Data Source Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.9.1, "An Attempt to Access a Remote 10.3.2 or Later WLS Data Source Fails"

- Section 2.9.2, "ORA-01591 Errors Occur on SOA Servers Configured to Use Multiple Oracle RAC Nodes"
- Section 2.9.3, "JDBC Connection Release is Slow Due to Non-XA Commit"

2.9.1 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 workarounds 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 workarounds:
  - 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 workarounds: (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 Oracle WebLogic 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 *Developing JTA Applications 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.

For additional information, see "Security Considerations for WebLogic RMI Drivers" in *Developing JDBC Applications for Oracle WebLogic Server*.

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

Platform: Linux

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.

2.9.3 JDBC Connection Release is Slow Due to Non-XA Commit

Platform: All

For a non-XA connection, `setAutoCommit(true)` is called if the connection is currently in auto-commit `false` state when a connection is closed. This should automatically commit any outstanding local transaction, but some drivers are not compliant with the JDBC specification and leave the transaction open. If the application does not complete (commit or rollback) the local transaction before closing the connection, the connection is returned to the pool with outstanding work that may never be completed or it may be committed or rolled back by the next reservation of that connection. To prevent this from happening, a WebLogic data source calls commit on the connection when returning it to the pool. See "Local Transaction Completion When Closing a Connection" in *Administering JDBC Data Sources for Oracle WebLogic Server*.

Workaround

This is a performance regression that can be fixed by setting the system property `weblogic.datasource.endLocalTxOnNonXaConWithCommit=false` if you know that your driver automatically commits the local transaction on `setAutoCommit(true)` or your application code always completes the transaction before calling close.

2.10 Deployment Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.10.6, "Munger:2156217 Error Occurs When Creating a Work Manager Component Using a Deployment Plan"

2.10.1 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.

2.10.2 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.

2.10.3 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();
```

2.10.4 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.

2.10.5 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.

2.10.6 Munger:2156217 Error Occurs When Creating a Work Manager Component Using a Deployment Plan

Platform: All

If you use the Administration Console to create a work manager component for a deployed application, WebLogic Server displays a `Munger:2156217` error when applying the deployment plan.

This error occurs under the following circumstances:

1. Deploy a Web application.
2. Use the Administration Console to open the application on the **Configuration > Workload** page.
3. Create a new work manager.
4. Open the work manager.
5. Create a new work manager component with type **Response Time Request Class**.
6. Click **Finish**.

   The page becomes empty and the `Munger:2156217` error displays in the Administration Console.

Workaround

Modify the deployment plan XML file and change the `xpath` element for the added work manager.

Change the `xpath` from:

```
<xpath>/weblogic-web-app/work-manager/[name="WorkManager-0"]/response-time-request
```

```
-class/[name="ResponseTimeReqClass-0"]/name</xpath>
```

to

```
<xpath>/weblogic-web-app/work-manager/[name="WorkManager-0"]/response-time-request
-class[name="ResponseTimeReqClass-0"]/name</xpath>
```

2.11 Developer Experience Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.11.1, "Users Need to Set BEA_HOME System Property While Using Appc For Pub-Sub Modules"
- Section 2.11.2, "weblogic-maven-plugin appc Goal's verbosejavac Parameter Case Is Incorrect"

2.11.1 Users Need to Set BEA_HOME System Property While Using Appc For Pub-Sub Modules

Platform: All

An error occurs when using the appc Maven plug-in after installing WebLogic Server Maven artifacts to the local repository using the Maven synchronization plug-in.

Workaround

WebLogic Server pub-sub libraries rely on the BEA_HOME system property to resolve compiler issues. Set the BEA_HOME system property while running appc on pub-sub applications for compilation to resolve these dependencies.

2.11.2 weblogic-maven-plugin appc Goal's verbosejavac Parameter Case Is Incorrect

Platform: All

In the `weblogic-maven-plugin appc` goal, the `verbosejavac` parameter has the incorrect case. It should be `verboseJavac` instead of `verbosejavac`. When you try to pass a value to the `verboseJavac` parameter from a command line with `-DverboseJavac=true`, it will not work since the expression is incorrectly `verbosejavac`.

Workaround

Use `-Dverbosejavac=true` from the command line.

2.12 EJB Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.12.1, "Primary Key in Oracle Table is CHAR"
- Section 2.12.2, "No Available Annotation That Enables Creation of a Clusterable Timer"
- Section 2.12.3, "Kodo's MappingTool Cannot Generate Schemas"
- Section 2.12.4, "Extensions to the JPA Metadata Model Can Only Be Specified Via Annotations"
- Section 2.12.5, "Lookup Method Injection Not Supported by Spring"

2.12.1 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.

2.12.2 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.

2.12.3 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.

2.12.4 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.

2.12.5 Lookup Method Injection Not Supported by Spring

Platform: All

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

2.12.6 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.

2.12.7 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.

2.12.8 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.

2.12.9 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.

2.12.10 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.

2.12.11 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();
}
```

2.12.12 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.

2.13 Examples Issues and Workarounds

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

2.14 HTTP Publish/Subscribe Server Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.14.1, "Authentication and Authorization of the Local Client is not Supported"

- Section 2.14.2, "Event Messages Published By Local Clients Do Not Go Through Filters"

2.14.1 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.

2.14.2 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.

2.15 Installation Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.15.1, "Installation Fails with Fatal Error"

2.15.1 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.

2.16 Java EE Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.16.1, "FastSwap May Relax the Access Modifiers of Fields and Methods"
- Section 2.16.2, "FastSwap Does Not Support Redefinition of the Entity Bean and ejbClass"
- Section 2.16.3, "Classpath Order Is Not Guaranteed When There Are Multiple JARs in an EAR File"
- Section 2.16.4, "FastSwap Not Supported When Using CDI"

2.16.1 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.

2.16.2 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.

2.16.3 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.

2.16.4 FastSwap Not Supported When Using CDI

Platform: All

FastSwap is not supported when using CDI. If you deploy an application in exploded format with FastSwap enabled, this deployment fails and errors related to CDI occur.

2.17 JDK Issues and Workarounds

This section describes the following issues and workarounds:

2.17.1 Oracle JRockit Not Supported For Execution of WebLogic Server 12.1.2 Server Applications

Platform: All

Oracle WebLogic Server 12.1.2 supports JDK 7 for execution of server applications, and JDK 6 and JDK 7 for WebLogic Server 12.1.2 clients connecting to WebLogic Server 12.1.2 servers. Oracle JRockit is not supported for execution of WebLogic Server 12.1.2 server applications. For more information, see "Supported Configuration" in *What's New in Oracle WebLogic Server*.

2.17.2 SSLv2Hello Obsolete in JDK 7

Platform: All

In versions of WebLogic Server prior to 12.1.2, if a client sent an SSLv2Hello, WebLogic Server converted it into an SSLv3Hello. However, WebLogic Server 12.1.2 supports only JDK 7 (clients can use JDK 6 or 7) and the SSLv2Hello protocol is not supported.

This means that a client that sends an SSLv2Hello will not be able to connect to a version 12.1.2 WebLogic Server. Clients must send SSLv3Hello.

Workaround

Set weblogic.security.SSL.protocolVersion=SSL3 to force a client to send SSLv3Hello.

2.18 JMS Issues and Workarounds

This section describes the following issues and workarounds:

2.18.1 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 J2EE specification-compliant module in the EAR.

2.18.2 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.

2.18.3 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.

2.18.4 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."

2.18.5 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`.

2.18.6 AQ JMS Dequeue Fails After Daylight Savings Change

Platform: All

When using ExadataDatabase 11.2.0.3, after Daylight Saving Time (DST) changes, when Oracle Advanced Queueing (AQ) JMS receives a message, dequeue fails with the following error: `JMS-120: Dequeue failed`, which is caused by `java.sql.SQLDataException: ORA-01878: specified field not found in datetime or interval`.

This occurs in some Message-Drive Bean (MDB) applications after DST changes and it may potentially happen for other AQ JMS applications that are receiving messages.

Workaround

Download and apply patch 13880758 for ExadataDatabase 11.2.0.3 to solve the AQ JMS issue in WebLogic Server. You can download this patch from My Oracle Support.

2.18.7 Custom Domain Template Upgrade May Result in Lost Topic Messages or Deplete Server Memory

Platform: All

As of WebLogic Server 12.1.2, JMS server and WebLogic store targeting in the Configuration Wizard has changed.

In 12.1.2, the Configuration Wizard automatically targets JMS servers and WebLogic stores to migratable targets when these objects are not explicitly targeted to a Managed Server or a cluster in a domain template. Using migratable targets is a best practice that enables high availability for the JMS system.

If you use a custom domain template to create domains in WebLogic Server 12.1.2, and that template includes JMS servers and WebLogic stores that are not explicitly targeted to a Managed Server or a cluster, targeting results will differ from previous releases.

This change in behavior also results in a change to durable topic subscriptions for message-driven beans (MDBs) that enable the `generate-unique-client-id` extension. When WebLogic Server creates durable topic subscriptions for such an MDB, it changes the subscription name to include the migratable target name. Messages stored under the original subscription names will not be delivered to the MDB, and the original subscriptions will continue to accumulate new messages.

When planning your upgrade, note the following important changes:

- If you follow the instructions in *Upgrading Oracle WebLogic Server* and use the Reconfiguration Wizard to reconfigure your existing pre-12.1.2 domain, the configuration and durable topic subscriptions will remain intact.
- If you regenerate your domain using a custom template, as described above, the resulting configuration will differ from previous releases and new durable topic subscriptions will be created when the system is started. However, old durable topic subscriptions will remain. Those subscriptions may contain unprocessed messages that will continue to accumulate messages, depleting server memory.

Workaround

Choose one of the following recommended workarounds:

- Use the Reconfiguration Wizard to upgrade the domain in place.
- Drain messages before upgrading or regenerating the domain configuration. After upgrading, use the WebLogic Server Administration Console to search and delete old JMS subscriptions.
- Delete JMS file store files or JMS JDBC store tables. Note that all messages persisted in the file or table will be deleted.

2.18.8 Configuration Wizard Allows Targeting of Bridges To Mixed and Dynamic Clusters

Platform: All

Targeting a messaging bridge to a mixed or dynamic cluster is not supported. No exception occurs when this is attempted.

2.18.9 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()
```

2.19 JNDI Issues and Workarounds

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

2.20 JTA Issues and Workarounds

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

2.21 Java Virtual Machine (JVM) Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.21.1, "1.4 Thin Client Applet Cannot Contact WebLogic Server"
- Section 2.21.2, "Applications Running on Some Processors May Experience Intermittent Time Issues"
- Section 2.21.3, "JRockit JVM Appears to Freeze When Doing Long Array Copies"

- Section 2.21.4, "Serial Version UID Mismatch"
- Section 2.21.5, "JVM Stack Overflow"
- Section 2.21.6, "Using AWT libraries May Cause a JVM Crash"
- Section 2.21.7, "Serial Version UID Mismatch"

2.21.1 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.

2.21.2 Applications Running on Some Processors May Experience Intermittent Time Issues

Platform: RedHat Linux

Applications that run on RedHat (RH) Linux 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. 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.

Please see `https://bugzilla.redhat.com/show_bug.cgi?id=452185`. This problem was fixed in RedHat 5.3.

2.21.3 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
```

2.21.4 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 *WL_HOME*/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.

2.21.5 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.

2.21.6 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
```

2.21.7 Serial Version UID Mismatch

Platform: Linux, AIX

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

Workaround

To be compatible with the serialization of previously compiled applications, modify the *WL_HOME*/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 appropriate Linux or AIX command line option.

AIX:

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

Linux:

```
export 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.

2.22 Monitoring Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.22.1, "MBean Attributes Not Explicitly Marked as @unharvestable Appear as Harvestable"
- Section 2.22.2, "Issue with Ambiguous Watch Rule ObjectName Patterns"
- Section 2.22.3, "HARVESTER_WM May Generate RuntimeException When Queue Capacity Exceeded"
- Section 2.22.4, "Deadlock May Occur in WebLogic Server Logging"

2.22.1 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`.

2.22.2 Issue with Ambiguous Watch Rule ObjectName Patterns

Platform: All

When specifying a wildcard pattern in a variable for a watch rule expression that matches custom MBean ObjectName patterns, ensure that the pattern is sufficiently explicit. If you exclude an MBean type name and use an ambiguous instance pattern, the following may result:

- Only WebLogic Server runtime MBean instances are matched to the pattern.
- The desired custom MBean instances are ignored.

For example, the following ObjectName pattern does not explicitly declare a type and uses an ambiguous ObjectName pattern that can match a WebLogic Server runtime MBean instance:

```
${ServerRuntime//com.b*:Type=Server*,*}
```

Workaround

To avoid confusion, use a sufficiently explicit ObjectName pattern, or declare the MBean type in the variable expression.

2.22.3 HARVESTER_WM May Generate RuntimeException When Queue Capacity Exceeded

Platform: All

The `HARVESTER_WM` is an internal Work Manager instance that is used to process MBean registration events in order to map new instances to (or remove them from) active harvester configurations as these instances are added or removed to the `ServerRuntime` MBean server.

The `HARVESTER_WM` may generate a `RuntimeException` when its queue capacity is being exceeded while running in certain JMS stress scenarios where connection/resource pooling is not used. In these scenarios, JMS creates and destroys several MBeans for every connection event and under heavy load these can number in the hundreds per second. This causes the harvester to process a high volume of MBean registration events. Because `HARVESTER_WM` has an unspecified `minThreadsConstraint`, the application or server may be under enough load to starve out the internal `HARVESTER_WM`, preventing it from being allocated a thread able to process these events. An application that also creates a high volume of JMX registration events could also lead to similar behavior.

These scenarios can be exacerbated by other contributing factors, including the amount of logging or debug messages the application or the server is configured to generate. These factors can add to the problem by slowing down and tying up system threads for longer periods of time, compound the starvation issue.

Workaround

There are several possible workarounds. You can disable any WLDF harvester configurations, including the WLDF built-in system resource, which is enabled by default in production. Another option is to trim down the amount of logging being performed, if applicable and/or possible. Finally, you can configure the application to use JMS connection pooling. For more information, see "Harvester Performance Considerations" in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

2.22.4 Deadlock May Occur in WebLogic Server Logging

Platform: All

A deadlock may occur in WebLogic Server under the following conditions:

- `stdout` is redirected to the server log.
- Server log rotation on startup is disabled.
- There is a high volume of logging to the server log/`stdout`.

Workaround

If WebLogic Server enters into a deadlock under the above conditions, the situation may be resolved by enabling log rotation on startup with the following in the server's command line and restarting the server:

```
-Dweblogic.log.RotateLogOnStartup=true
```

It may also be configured by setting the `RotateLogOnStartup` property in the server's `LogMBean`. This can be done with WLST as follows:

```
edit()
startEdit()
cd('/Servers/yourserver/Log/yourserver')
```

```
cmo.setRotateLogOnStartup(true)
save()
activate()
```

Note: replace 'yourserver' with the name of your Managed Server.

2.23 Node Manager Issues and Workarounds

This section describes the following Node Manager issue and workaround:

- Section 2.23.1, "Removing Primary Interface Causes Error During Server Migration"
- Section 2.23.2, "Node Manager Not Putting Up -D64 When Starting Server Using Java Command"
- Section 2.23.3, "Oracle HTTP Server Instances Start in UNKNOWN State"

2.23.1 Removing Primary Interface Causes Error During Server Migration

Platform: Linux

On some specific Linux platforms and versions, there is an issue removing a virtual interface/alias dynamically. Removing the virtual interface that is the primary address of the interface may result in other secondary virtual IP addresses being removed at the same time. This may lead to random exceptions occurring with Node Manager during server migration. If you have this issue, you may occasionally find exceptions in the Node Manager log file when shutting down a server after migration. For example, you may receive the following error:

```
java.io.IOException: Command '/<PATH to DOMAIN>/bin/server_
migration/wlsifconfig.sh -removeif -IPv4 eth0 X.X.X.X returned an
unsuccessful exit code '1'.
```

Here is an example of the issue:

First, add three virtual interfaces, with the first one being the primary:

```
$ sudo /sbin/ifconfig eth0:4 X.X.X.178 netmask 255.255.248.0
$ sudo /sbin/ifconfig eth0:5 X.X.X.179 netmask 255.255.248.0
$ sudo /sbin/ifconfig eth0:6 X.X.X.180 netmask 255.255.248.0
$ sudo /sbin/ifconfig eth0:4 down
```

When removing the primary (first one in list), the other two will be automatically removed at the same time.

Workaround

To fix this issue temporarily, use the following command to enable the `promote_secondaries` flag on your network interface. Replace `eth0` with your actual interface name:

```
$ sudo /sbin/sysctl net.ipv4.conf.eth0.promote_secondaries=1
```

You can also use the following command to update the default setting for all interfaces:

```
$ sudo /sbin/sysctl net.ipv4.conf.all.promote_secondaries=1
```

If this is enabled and the primary address of an interface gets deleted, a secondary interface will be upgraded to become the primary interface. The default is to purge all the secondary interfaces when you delete the primary interface.

To permanently remedy this issue after server reboot, update the `sysctl.conf` file. For example:

```
$ echo "net.ipv4.conf.eth0.promote_secondaries=1" >> /etc/sysctl.conf
```

2.23.2 Node Manager Not Putting Up -D64 When Starting Server Using Java Command

Platform: Solaris on x86-64, SPARC 64, and HPUX IA64

There is a fundamental difference between Node Manager starting a server using a start script and Node Manager starting a server using the Java command. When using the start script, the -d64 flag is added based on some script language that detects the platform. Node Manager does not add this flag when starting a server with the Java command.

Workaround

When starting a server using Node Manager through the Java command, specify arguments such as -d64 in the `ServerStart` arguments field.

2.23.3 Oracle HTTP Server Instances Start in UNKNOWN State

Platform: All

In rare cases, Oracle HTTP Server (OHS) instances that are managed by WebLogic Server may start in state `UNKNOWN`. This can occur if the Administration Server is unable to initialize the state of the OHS instance, for example, if Node Manager is not running at the time the OHS instance is created and if you connect directly to Node Manager and bypass the Administration Server when checking the state for the first time.

Workaround

Continue to use the Administration Server. The state of the OHS instance should be properly initialized.

2.24 Operations, Administration, and Management Issues and Workarounds

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

2.25 Oracle Kodo Issues and Workarounds

This section describes the following Oracle Kodo issue and workaround:

- Section 2.25.1, "Value Retrieved for an Empty Byte Array Field is NULL"

2.25.1 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.

2.26 Plug-ins Issues and Workarounds

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

- Section 2.26.1, "apr_socket_connection Exception Occurs When Using the IIS Plug-In"
- Section 2.26.2, "Failure to Introspect Write Protected Domains With Managed Servers"
- Section 2.26.3, "SYSPROP Enables HTTP Proxying in OVAB Studio"

2.26.1 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.

2.26.2 Failure to Introspect Write Protected Domains With Managed Servers

Platform: All

Introspection fails and users receive an error when they try to introspect a domain that they cannot write into.

Workaround

Change the permissions on the domain root directory to allow the user that executes the introspect.

2.26.3 SYSPROP Enables HTTP Proxying in OVAB Studio

Platform: All

In Oracle Virtual Assembly Builder (OVAB) Studio, HTTP proxying is disabled. You can use a system property to enable HTTP proxy detection.

You can set this system property for each execution of a Studio launch, or permanently by modifying the `abstudio.sh` file.

To set the property for a single execution of OVAB Studio:

1. Shut down OVAB Studio.
2. Remove the configuration directory:

   `$AB_INSTANCE/state/gui/$USER/system.12.1.2.0.0` (or equivalent)
3. Restart the GUI with the property set to some value, for example 1:

```
./abstudio.sh -J-Dovab.studio.enableHttpProxy=1
```

You must define the property in every ensuing execution of the GUI or the property setting in `abstudio.sh` will force proxying back to false.

To set the property to consistently enable HTTP proxying:

1. Edit the `abstudio.sh` file in the instance bin directory.
2. Add the property setting to SYSPROPS as follows:

```
SYSPROPS="${SYSPROPS} -J-Dovab.studio.enableHttpProxy=1
```

After setting `enableHTTPProxy=1`, you can set the proxy host, port, and exceptions using the standard Java properties `http.proxyHost`, `http.proxyPort`, and `http.nonProxyHosts`. If you are using a nonstandard desktop environment on Linux, you may need to set the `http_proxy` property with the `valuehost:port`.

2.27 Protocols Issues and Workarounds

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

2.28 RMI-IIOP Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.28.1, "Ant 1.7 rmic Task Incompatibility"
- Section 2.28.2, "Truncated Java Exception Stack Trace Returned to Client if EJB Invocation Fails"

2.28.1 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}"
```

2.28.2 Truncated Java Exception Stack Trace Returned to Client if EJB Invocation Fails

Platform: All

When a client invokes an EJB that is hosted in a WebLogic 12.1.2 domain configured to run in production mode, any invocation failure results in a truncated Java exception stack trace returned to the client.

Workaround

In the Java command that starts WebLogic Server, specify the following option:

```
-Dweblogic.PrintStackTraceInProduction=true
```

2.29 Security Issues and Workarounds

This section describes the following issues and workarounds:

2.29.1 Service-side Kerberos Authentication Fails With Error 401

Platform: All

Service-side Kerberos authentication fails with an HTTP/1.1 `Error 401 Unauthorized` if the JDK version is JRockit 1.60_24, 1.60_28, or 1.60_29.

Workaround

Use one of the following workarounds:

- Instead of using `ktab.exe` to generate the keytab file, use another tool such as `kadmin` to generate it.
- Use `ktab.exe` to manually supply the correct kvno.

2.29.2 BAD_MAC_RECORD Error Occurs When Using JSSE-based SSL Provider

Platform: All

If WebLogic Server is configured to use the JSSE-based SSL provider, attempts to create an SSL connection may fail with a `BAD_MAC_ERROR` message.

Workaround

Install JDK 7u2 or higher and restart WebLogic Server.

2.29.3 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.

2.29.4 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.

2.29.5 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.

2.29.6 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.

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

2.29.8 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 administrator 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.

2.29.9 RCU Schema Creation Fails with IBM DB2 for WebLogic Server Components

Platform: All

The Oracle Fusion Middleware Repository Creation Utility (RCU) may fail to execute the WebLogic Server security store scripts when the target database is IBM DB2.

Workaround

Edit the relevant `$MW_HOME/wlserver/server/lib/rdbms_security_store_db2.sql` file by replacing each semicolon (`;`) with the "`@`" character.

2.29.10 RCU Schema Creation Fails with Microsoft SQL Server for WebLogic Server Components

Platform: All

The Oracle Fusion Middleware Repository Creation Utility (RCU) may fail to execute the WebLogic Server security store scripts when the target database is Microsoft SQL Server.

Workaround

Edit the relevant `$MW_HOME/wlserver/server/lib/rdbms_security_store_sqlserver.sql` file by adding the word "`GO`" after each SQL statement.

2.30 SNMP Issues and Workarounds

This section describes the following issues and workarounds:

2.30.1 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*.

2.31 Spring Framework on WebLogic Server Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.31.1, "OpenJPA ClassFileTranformer Does Not Work When Running on JRockit"
- Section 2.31.2, "petclinic.ear Does Not Deploy on WebLogic Server"

2.31.1 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`.

2.31.2 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.

2.32 System Component Architecture (SCA) Issues and Workarounds

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

2.33 Upgrade Issues and Workarounds

This section describes the following issue:

- Section 2.33.1, "Domains Created on WebLogic Server 10.3.1 Cannot Be Run on WebLogic Server 10.3"

2.33.1 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.

2.34 Web Applications Issues and Workarounds

This section describes the following issues and workarounds:

2.34.1 MaxPostSizeExceededException Reported in Web Browser

After upgrading an application from a WebLogic Server version prior to 12.1.2, a `MaxPostSizeExceededException` is reported in the web browser.

Workaround

Set the `max-save-post-size` session-descriptor to the maximum size (in bytes) of the POST that will be saved or buffered by the container during FORM authentication.

2.34.2 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.

2.34.3 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.

2.34.4 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.

2.34.5 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.

2.34.6 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/.

2.34.7 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.

2.34.8 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.

2.34.9 503 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.

2.34.10 Applications Configuring jdbc-connection-timeout-secs Fail to Deploy

Platform: All

As of WebLogic Server 12.1.2, the `jdbc-connection-timeout-secs` element in the `weblogic.xml` deployment descriptor has been removed. Applications that configure `jdbc-connection-timeout-secs` will fail to deploy on WebLogic Server 12.1.2 server instances, resulting in the following error in the server log:

```
Unable to load descriptor /.../WEB-INF/weblogic.xml of module myweb. The error is
weblogic.descriptor.DescriptorException: VALIDATION PROBLEMS WERE FOUND
  <6:7> problem: cvc-complex-type.2.4a: Expected elements
'timeout-secs@http://xmlns.oracle.com/weblogic/weblogic-web-app ...' instead of
'jdbc-connection-timeout-secs@http://xmlns.oracle.com/weblogic/weblogic-web-app'
here in element
```

```
session-descriptor@http://xmlns.oracle.com/weblogic/weblogic-web-app
```

Workaround

Remove the `jdbc-connection-timeout-secs` element from the `weblogic.xml` deployment descriptor.

2.34.11 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.

2.35 WebLogic Server Scripting Tool (WLST) Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.35.1, "Permission Denied Error Occurs for WLST Offline Logging"
- Section 2.35.2, "Property Names Containing '.' Characters Are Not Supported by loadProperties"
- Section 2.35.3, "Invalid cachedir Created by Jython Causes WLST to Error Out"
- Section 2.35.4, "Domain Creation Fails on Windows Server 2012"

2.35.1 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.

2.35.2 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:

http://docs.python.org/tut/node8.html

2.35.3 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`.

2.35.4 Domain Creation Fails on Windows Server 2012

Platform: All

Domain creation with WLST fails when running WebLogic Server 12.1.2 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. Log in to My Oracle Support at `http://support.oracle.com`. Select the **Patches and Updates** page and enter the patch number, 16844206, in the search field. Select **Patch 16844206: SU Patch [NPM3]: WLST CANNOT GET ENV ON WINDOWS SERVER 12 WITH MINIMAL ENV** and select **WLS 12.1.2.0** from the **Release** dropdown menu on the Patch Details page.

2.36 Web Server Plug-Ins Issues and Workarounds

This section describes the following issue:

- Section 2.36.1, "MOD_WLS_OHS Does Not Fail Over"

2.36.1 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.

2.37 Web Services and XML Issues and Workarounds

This section describes the following issues and workarounds:

- Section 2.37.1, "Exceptions Occur When Using Asynchronous MakeConnection in a Clustered Environment"
- Section 2.37.2, "weblogic.wsee.jaxws.mdb.DispatchPolicy WorkManager Cannot Be Found"
- Section 2.37.3, "Troubleshooting Problems When Applying the WebLogic Advanced Web Services for JAX-WS Extension Template"
- Section 2.37.4, "Sparse Arrays and Partially Transmitted Arrays Are Not Supported"
- Section 2.37.5, "WSDL Compiler Does Not Generate Serializable Data Types"
- Section 2.37.6, "Use of Custom Exception on a Callback"

2.37.1 Exceptions Occur When Using Asynchronous MakeConnection in a Clustered Environment

When using an asynchronous `MakeConnection` in a clustered environment, one of the following exceptions occurs:

- `McPollingTimeoutException`
- `IllegalStateException`

There is no workaround for these issues.

2.37.2 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.

2.37.3 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 *Developing JAX-WS Web Services for Oracle WebLogic Server*.

2.37.4 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.

2.37.5 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.

2.37.6 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.

2.37.7 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.

2.37.8 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.

2.37.9 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.

2.37.10 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.

2.37.11 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.

2.37.12 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.

2.37.13 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`.

2.37.14 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 {
  ...
}
```

2.37.15 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/>`.

2.37.16 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.

2.37.17 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.

2.37.18 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.

2.37.19 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.

2.37.20 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.

2.37.21 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 *Tuning Performance of Oracle WebLogic Server*.

2.37.22 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.

2.38 WebLogic Tuxedo Connector Issues and Workarounds

This section describes the following issue and workaround:

- Section 2.38.1, "View Classes are not Set on a Per Connection Basis"

2.38.1 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.

2.39 Documentation Errata

This section describes documentation errata:

- Section 2.39.1, "Configuration Wizard Domain Creation Options Are Not Available"
- Section 2.39.2, "Issues With Search Function in the Samples Viewer"
- Section 2.39.3, "Japanese Text Displays in Some Search Results Topics Avitek Medical Records"
- Section 2.39.4, "HTML Pages For Downloaded Libraries Do Not Display Properly"
- Section 2.39.5, "Documentation Update For Partial Redeployment Behavior"
- Section 2.39.6, "OPSS Keystore Service Supported For WebLogic Web Service SSL"
- Section 2.39.7, "Online Help For Configuring the RDBMS Security Store is Incorrect"
- Section 2.39.8, "Documentation Update Related to JSF Implementation"
- Section 2.39.9, "Documentation Uses Incorrect Java EE Version Names"

2.39.1 Configuration Wizard Domain Creation Options Are Not Available

Due to a late change in the Fusion Middleware Configuration Wizard, the Configuration Wizard help describes two domain creation options that are not available. The Create Expanded Domain and Create Compact Domain options were removed. Compact domains can be created only by using the WebLogic Scripting Tool (WLST).

For the most up-to-date documentation on the Fusion Middleware Configuration Wizard, see http://docs.oracle.com/docs/cd/E24329_01/index.htm and reference *Creating Domains Using the Configuration Wizard*.

2.39.2 Issues With Search Function in the Samples Viewer

Platform: All

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**.

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

Platform: All

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

2.39.4 HTML Pages For Downloaded Libraries Do Not Display Properly

Platform: All

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.

2.39.5 Documentation Update For Partial Redeployment Behavior

The documentation is incorrect for the recommended usage for partial redeployment of graphics files, static HTML files, and JSPs. The current documentation states that this behavior is safe for all production environments. However, the description for this recommended usage should read as follows:

Use only during scheduled application downtime, or when it is not critical to preserve client connections and in-process work.

2.39.6 OPSS Keystore Service Supported For WebLogic Web Service SSL

The documentation incorrectly states that the OPSS Keystore Service is not supported.

As described in "Configuring Oracle OPSS Keystore Service" in *Administering Security for Oracle WebLogic Server*, the OPSS Keystore Service provides an alternate mechanism to manage keys and certificates.

> **Note:** You can use the OPSS Keystore Service only if you have installed the Oracle JRF template on the WebLogic Server system and used this template to create the domain. The OPSS Keystore Service is available only with the JRF template and is not available with the default WebLogic Server configuration.

WebLogic Web services policies that require Secure Sockets Layer (SSL) can use an existing OPSS Keystore Service without additional configuration.

However, WebLogic Web service policies that use keys and certificates for message protection, as described in "Configuring Message-Level Security" in *Securing WebLogic Web Services for Oracle WebLogic Server* cannot use the OPSS Keystore Service. Specifically, do not configure the `IntegrityKeyStore` and `ConfidentialityKeyStore` properties to use an OPSS Keystore Service.

Oracle recommends that you instead use OWSM message protection policies with the OPSS Keystore Service, as described in "Configuring Message Protection" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

2.39.7 Online Help For Configuring the RDBMS Security Store is Incorrect

Platform: All

As of WebLogic Server 12.1.2, the Configuration Wizard no longer supports configuring the RDBMS security store. To configure the RDBMS security store, you must use WLST. The following Administration Console online help incorrectly describes using the Configuration Wizard to configure the RDBMS security store:

- "Create a new domain to use the RDBMS security store"
- "Upgrade an existing domain to use the RDBMS security store"

Workaround

For information on configuring the RDBMS security store using WLST, see "Managing the RDBMS Security Store" in *Administering Security for Oracle WebLogic Server*.

2.39.8 Documentation Update Related to JSF Implementation

Platform: All

A previous version of the documentation in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server* incorrectly indicated that you can override the WebLogic Server JSF version by bundling a different version of JSF with your application. If you encounter any issues with your JSF implementation, Oracle recommends visiting My Oracle Support at http://support.oracle.com/ and filing a support issue.

2.39.9 Documentation Uses Incorrect Java EE Version Names

Platform: All

The WebLogic Server Administration Console online help in this release contains some references to J2EE and J2SE. These references should instead be to Java EE and Java SE, respectively. For more information on the Java version naming updates, see http://www.oracle.com/technetwork/java/javase/overview/javanaming-2227065.html.