

Oracle® Fusion Middleware

What's New in Oracle WebLogic Server

11g Release 1 (10.3.4)

E13852-05

January 2011

Welcome to Oracle WebLogic Server. The following sections describe new and changed functionality in this WebLogic Server release.

Note: Oracle Fusion Middleware 11g contains Oracle WebLogic Server 11g. The current version number of Oracle WebLogic Server is 10.3.4.

- [Section 1, "Core Server"](#)
- [Section 2, "Deployment"](#)
- [Section 3, "WebLogic Diagnostic Framework"](#)
- [Section 4, "Enterprise Java Beans \(EJBs\)"](#)
- [Section 5, "Installation and Upgrade"](#)
- [Section 6, "JDBC"](#)
- [Section 7, "JTA"](#)
- [Section 8, "Messaging"](#)
- [Section 9, "Miscellaneous"](#)
- [Section 10, "Security"](#)
- [Section 11, "WebLogic Service Component Architecture \(WebLogic SCA\)"](#)
- [Section 12, "Web Applications, Servlets, and JSPs"](#)
- [Section 13, "Web Services"](#)
- [Section 14, "Deprecated Functionality \(WebLogic Server 11g Release 1\)"](#)
- [Section 15, "Deprecated Functionality \(WebLogic Server 10.3\)"](#)
- [Section 16, "Standards Support"](#)
- [Section 17, "Supported Configurations"](#)
- [Section 18, "Documentation Accessibility"](#)

1 Core Server

This section describes the new Core Server features for WebLogic Server 10.3.4.

1.1 Exalogic Elastic Cloud Software Support

The following performance optimizations have been added to WebLogic Server to support Oracle Exalogic Elastic Cloud Software:

- You can enable the `DomainMBean.ExalogicOptimizationsEnabled` MBean attribute, which has been added to WebLogic Server to provide thread management, request processing, and reduced lock contention optimizations used for Oracle Exalogic. Enabling `DomainMBean.ExalogicOptimizationsEnabled`, in turn, enables the following MBean attributes that have also been added to support Oracle Exalogic:
 - The `ScatteredReadsEnabled` and `GatheredWritesEnabled` attributes of the `KernelMBean`, which increase efficiency during I/O in environments with high network throughput. Alternatively, these optimizations can be enabled by the following WebLogic Server startup command options:
- | KernelMBean Attribute | Corresponding WebLogic Server Startup Option |
|------------------------------------|--|
| <code>ScatteredReadsEnabled</code> | <code>-Dweblogic.ScatteredReadsEnabled=true</code> |
| <code>GatheredWritesEnabled</code> | <code>-Dweblogic.GatheredWritesEnabled=true</code> |
- The `ClusterMBean.SessionLazyDeserializationEnabled` attribute, which increases efficiency with session replication. Alternatively, this optimization can be enabled by the `-Dweblogic.replication.enableLazyDeserialization=true` WebLogic Server startup command option.
 - The `KernelMBean.AddWorkManagerThreadsByCpuCount` attribute, which increases efficiency of the self-tuning thread pool by aligning it with the Exalogic processor architecture threading capabilities.
- You can configure a network channel to use socket direct protocol (SDP). See "Enable Socket Direct Protocol for a network channel" in *Oracle WebLogic Server Administration Console Help*.

Note: Usage of these features may require additional configuration. For more information, see *Oracle Fusion Middleware Licensing Information*.

For information about using these optimizations for Oracle Exalogic Elastic Cloud Software, see "Enabling Exalogic-Specific Enhancements in Oracle WebLogic Server 11g Release 1 (10.3.4)" in *Exalogic Elastic Cloud Enterprise Deployment Guide*.

1.2 Tuning Cached Connections

This release provides the `http.keepAliveCache.socketHealthCheckTimeout` property for tuning the behavior of returning a socket connection from the cache when keep-alive is enabled in the HTTP 1.1 protocol. For more information, see "Tuning Cached Connections" in *Performance and Tuning for Oracle WebLogic Server*.

2 Deployment

This section describes the new deployment features for WebLogic Server 10.3.4.

- [Apache Maven Plug-in](#)
- [weblogic.deploy.UploadLargeFile Property](#)

2.1 Apache Maven Plug-in

Now you can use an Apache Maven plug-in for WebLogic Server (`weblogic-maven-plugin`) to perform deployment operations similar to those supported by the command-line utility, `weblogic.Deployer`. The plug-in lets you deploy, redeploy, update, and such, applications built using Maven to WebLogic Server from within the Maven environment. For more information, see "Using the WebLogic Maven Plug-In for Deployment" in *Deploying Applications to Oracle WebLogic Server*.

2.2 weblogic.deploy.UploadLargeFile Property

A new system property, `weblogic.deploy.UploadLargeFile`, has been added to prevent an issue that may occur when a large application file is deployed using the `upload` option. Include this flag in the `java` command you use to launch a deployment client if you receive a `java.lang.OutOfMemoryError: Java heap space error`.

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.

3 WebLogic Diagnostic Framework

In version 10.3.3, the WebLogic Diagnostic Framework introduced the WLDF diagnostic volume setting, which controls the amount of data that is automatically produced by WebLogic Server at run time and captured in the JRockit Flight Recorder file. By default, the WLDF diagnostic volume is set to `Off` in version 10.3.3. However, in version 10.3.4, the default setting is changed to `Low`.

For information about the WLDF diagnostic volume setting and using it to propagate WebLogic Server events to JRockit Flight Recorder, see "Configuring Diagnostic Image Capture for JRockit Flight Recorder" in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*. For general information about WLDF integration features with JRockit Flight Recorder, see "Using WLDF with Oracle JRockit Flight Recorder."

4 Enterprise Java Beans (EJBs)

This section describes the new EJB features for WebLogic Server 10.3.4.

- [New Message-Driven Bean \(MDB\) Activation Configuration Properties](#)
- [Enhanced Support for JMS Distributed Topics in Message-Driven Beans \(MDB\)](#)
- [Support for JPA 2.0 Providers](#)

4.1 New Message-Driven Bean (MDB) Activation Configuration Properties

WebLogic Server 10.3.4 includes new Message-Driven Bean (MDB) activation configuration properties, configuration permutations, and the deployment actions that provide high availability and parallel processing. See "Deployment Elements and Annotations for MDBs" and "Configuring and Deploying MDBs Using JMS Topics" in *Programming Message-Driven Beans for Oracle WebLogic Server*.

4.2 Enhanced Support for JMS Distributed Topics in Message-Driven Beans (MDB)

MDB can take advantage of new JMS features which provide more scalability and high availability features for distributed topics (DT).

In addition to the new activation configuration properties described above, WebLogic Server 10.3.4 provides enhanced support for JMS distributed topics. MDB deployments can now be more scalable and can provide higher availability when working with distributed topics, especially when using durable subscriptions. See *Programming Message-Driven Beans for Oracle WebLogic Server*, especially the following chapters:

- "Configuring and Deploying MDBs Using JMS Topics"
- "Deployment Elements and Annotations for MDBs"
- "Topic Deployment Scenarios"
- "Topic Subscription Identifiers"
- "How WebLogic MDBs Leverage WebLogic JMS"

4.3 Support for JPA 2.0 Providers

An available patch for WebLogic Server 10.3.4 enables support for TopLink, a Java Persistence Architecture (JPA) 2.0 provider. JPA 2.0 includes improvements and enhancements to domain modeling, object/relational mapping, EntityManager and Query interfaces, and the Java Persistence Query Language (JPQL). It includes a new API for criteria queries, a metamodel API, and support for validation. For the JPA 2.0 Specification, see "JSR-000317 Java Persistence 2.0 (Final Release)" at <http://jcp.org/aboutJava/communityprocess/final/jsr317/index.html>.

To enable JPA 2.0 support, you must download and install a WebLogic Server patch. JPA 2.0 is not supported on the openJPA/Kodo that is shipped with WebLogic Server 10.3.4.

- Log in to My Oracle Support and download patch 9923849.
- If using Smart Update, download and apply public patch QWG8.

Note: Support for JPA 2.0 in WebLogic Server is provided as a patch because JPA 2.0 is part of Java Platform, Enterprise Edition (Java EE) 6. Therefore, enabling JPA 2.0 support in the current release results in WebLogic Server not meeting all Java EE 5 compatibility requirements. To maintain Java EE 5 compatibility in the current release, the files required for the support are not enabled by default, although they are included in a standard WebLogic Server installation.

For more information, see "Using JPA 2.0 With TopLink in WebLogic Server" in *Programming WebLogic Enterprise JavaBeans, Version 3.0 for Oracle WebLogic Server*.

5 Installation and Upgrade

This section describes the new installation and upgrade features for WebLogic Server 10.3.4.

- [WebLogic Server Installer for Linux 64-Bit Operating System](#)
- [Supplement ZIP Installer for Development-Only Installations](#)

5.1 WebLogic Server Installer for Linux 64-Bit Operating System

A WebLogic Server installer for Linux x86 64-bit (`wls1034_linux64.bin`) is available for WebLogic Server 10.3.4. This installer includes WebLogic Server, Coherence, and Oracle JRockit 1.6.0.

5.2 Supplement ZIP Installer for Development-Only Installations

As of this release, Oracle provides a supplement ZIP installer for development-only installations. This installer supplements the development-only installer that has been provided since WebLogic Server 10.3.3. The supplemental installer includes the following additional WebLogic Server components for development-only environments:

- Samples files (extracted to `WL_HOME/samples`)
- Derby database and relevant scripts (extracted to `WL_HOME/common/derby`)
- Administration Console help files for non-English locales (extracted to `WL_HOME/server/lib/consolaapp/consolehelp/locale_directory`), where the locale directories are:
 - de
 - es
 - fr
 - it
 - ja
 - ko
 - pt-br
 - zh-cn
 - zh-tw

For more information, see "Development-Only Installation" in *Oracle WebLogic Server Installation Guide*.

6 JDBC

This section lists new JDBC features for WebLogic Server 10.3.4:

- [Enhanced Oracle RAC Support](#)
- [Reorganization of JDBC Resources](#)
- [Data Source Factories](#)
- [Socket Direct Protocol](#)
- [Connection-based System Properties](#)
- [Advanced Driver Properties](#)
- [Keep Connection After Local Transaction](#)

- [getVendorConnectionSafe](#)
- [Change in Data Direct Error Message Prefix](#)

6.1 Enhanced Oracle RAC Support

This release provides a new data source type, a Grid Link Data Source, to provide enhanced support for Oracle RAC. Grid Link Data Sources provide:

- A simplified and consolidated method to configure Oracle RAC connectivity between a single WebLogic Server data source and a service targeted to an Oracle RAC cluster.
- The ability to adaptively respond to state changes in the Oracle RAC Cluster, including support for:
 - Fast Connection Failover (FCF)
 - Runtime Connection Load Balancing (RCLB)
 - Graceful data base shutdown
- Additional performance features, such as XA Affinity for global transactions, that ensures all the data base operations performed on an Oracle RAC cluster are directed to the same Oracle RAC instance.
- Supports for Oracle Single Client Access Name (SCAN) addresses. This allows multiple Grid Link data source database listener addresses and ONS daemon addresses to register to SCAN listeners and appear as a single name used to access an Oracle RAC instance. A Grid Link data source containing SCAN addresses does not need to change if you add or remove Oracle RAC nodes.

For more information, see "Using Grid Link Data Sources" in *Configuring and Managing JDBC for Oracle WebLogic Server*.

6.2 Reorganization of JDBC Resources

JDBC resources have been reorganized within the WebLogic Server Administration Console domain tree. Generic Data Sources, Grid Link Data Sources, and Multi Data Sources are all located under the **Services > Data Sources** branch of the Domain Tree in the Administration Console.

6.3 Data Source Factories

Data Source Factories (deprecated) are still supported in this release but do not appear in the domain tree unless you have upgraded an existing domain that contains Data Source Factories. If your upgraded domain contains Data Source Factories, they are located under the **Services > Data Source Factories** branch of the Domain Tree in the Administration Console.

6.4 Socket Direct Protocol

This release supports Socket Direct Protocol (SDP), a high-performing socket implementation. If your environment supports SDP, see "Enable Socket Direct Protocol for a Grid Link Data Source" in the *Configuring and Managing JDBC for Oracle WebLogic Server*.

6.5 Connection-based System Properties

This release supports setting driver properties using the value of system properties. The value of each property is derived at runtime from the named system property. You can configure connection-based system properties using the Administration Console by editing the `System Properties` attribute of your data source configuration.

6.6 Advanced Driver Properties

WebLogic Server provides several attributes that provide improved Data Source performance when using Oracle drivers, including:

- Oracle JDBC JavaNet Fastpath: Enables the Oracle JDBC JavaNet Fastpath to reduce data copies and fragmentation.
- Oracle Optimize UTF8 Conversion: Enables the Oracle JDBC optimize UTF-8 conversion option.

If your environment supports these attributes, see "Configure Oracle parameters" in the *Oracle WebLogic Server Administration Console Help*.

6.7 Keep Connection After Local Transaction

This release provides a new attribute, `Keep Connection After Local Transaction`, that enables WebLogic Server to keep a physical database connection associated with a logical connection when committing or rolling back a local transaction. See "KeepConnAfterLocalTx" in the *Oracle WebLogic Server MBean Reference*.

6.8 getVendorConnectionSafe

This release provides `getVendorConnectionSafe` to access a physical connection. Similar to `getVendorConnection`, this mechanism returns the underlying physical connection (the vendor connection) from a pooled database connection (a logical connection). However, when the connection is closed, it is returned to the pool, independent of the setting of `Remove Infected Connections Enabled`. For some applications, `getVendorConnectionSafe` may enhance performance by eliminating the excessive creation of connections. For more information, see "Getting a Physical Connection from a Data Source" in *Programming JDBC for Oracle WebLogic Server*.

6.9 Change in Data Direct Error Message Prefix

In this release, error messages generated when using Data Direct database drivers now have the prefix of FMWGEN instead of OWLS.

7 JTA

This section describes the following new and changed WebLogic JTA features in this release of Oracle WebLogic:

- [Read-only, One-Phase Commit Optimizations](#)
- [New Committed Transaction Monitoring Statistics](#)

7.1 Read-only, One-Phase Commit Optimizations

When resource managers, such as the Oracle Database (including Oracle AQ and Oracle RAC), provide read-only optimizations, Oracle WebLogic can provide a read-only, one-phase commit optimization that provides a number of benefits—even when enabling multiple connections of the same XA transactions—such as eliminating `XAResource.prepare` network calls and transaction log writes, both in Oracle WebLogic and in the resource manager.

Note: Read-only, One-phase Commit Optimization requires Oracle DB 11.1.0.7.3PSU or above.

See "Read-only, One-Phase Commit Optimizations" in *Programming JTA for Oracle WebLogic Server*.

7.2 New Committed Transaction Monitoring Statistics

This release provides new committed transaction monitoring statistics, which together break down the Transaction Committed Total Count statistic:

- **Transaction No Resources Committed Total Count**—The total number of transactions with no enlisted resources that were committed since the server was started.
- **Transaction One Resource One Phase Committed Total Count**—The total number of transactions with only one enlisted resource that were one-phase committed since the server was started.
- **Transaction Read Only One Phase Committed Total Count**—The total number of transactions with more than one enlisted resource that were one-phase committed due to read-only optimization since the server was started.
- **Transaction Two Phase Committed Total Count**—The total number of transactions with more than one enlisted resource that were two-phase committed since the server was started.
- **Transaction LLR Committed Total Count**—The total number of LLR transactions that were committed since the server was started.

See "Monitoring Read-only, One-phase Transaction Statistics" in *Programming JTA for Oracle WebLogic Server*.

8 Messaging

This release includes the following new and changed features:

- Weighted Distributed Destinations are deprecated in WebLogic Server 10.3.4.0. Oracle recommends using Uniform Distributed Destinations.
- Partitioned Distributed Topics provide the ability to load balance messages to members which provides a highly scalable and available publishing mechanism. See "Configuring Partitioned Distributed Topics" in *Configuring and Managing JMS for Oracle WebLogic Server*.
- Unrestricted Client ID Policy allows more than one JMS connection can use the same Client ID. See "Configure an Unrestricted ClientID" in *Configuring and Managing JMS for Oracle WebLogic Server*.

- Shared Subscriptions allow multiple subscribers to share the same subscription which enables parallel processing of messages of a single subscription. See "Configure Shared Subscriptions" in *Configuring and Managing JMS for Oracle WebLogic Server*.
- New Message-Driven Bean (MDB) activation configuration properties, configuration permutations, and the deployment actions that provide high availability and parallel processing. See "Strategies for Configuring and Deploying MDBs Using Distributed Topics" in *Programming Message-Driven Beans for Oracle WebLogic Server*.
- Advanced WebLogic JMS publish and subscribe (pub/sub) concepts and functionality of Uniform Distributed Topics (UDTs) necessary to design high availability applications. See "Developing Advanced Pub/Sub Applications" in *Programming JMS for Oracle WebLogic Server*.
- The JMSDestinationAvailabilityHelper API provides a means for getting notifications when destinations become available or unavailable. These APIs are for advanced use cases only. Use this helper only when standard approaches for solving WebLogic distributed consumer problems have been exhausted. See "Advanced Programming with Distributed Destinations Using the JMS Destination Availability Helper API" in *Programming JMS for Oracle WebLogic Server*.

9 Miscellaneous

This section lists and describes miscellaneous changes in WebLogic Server 10.3.4.

9.1 WorkShop for WebLogic Is Not Supported

As of WebLogic Server 10.3.4, WorkShop for WebLogic is no longer supported. Oracle recommends that you use Oracle JDeveloper instead.

Prior to upgrading to WebLogic Server 10.3.4, you must remove WorkShop for WebLogic from your previous WebLogic Server installation. WebLogic Server 10.3.4 does not support removal of WorkShop components.

10 Security

This section lists and describes the following new features that have been added to the WebLogic Security Service for version 10.3.4:

- [RSA Cryptographic Services Upgrade](#)
- [Validation of Security Certificates](#)
- [SAML Single Sign-On Attribute Support](#)

10.1 RSA Cryptographic Services Upgrade

This release of WebLogic Server supports the RSA cryptographic services in JDK 6. For information about SSL support in WebLogic Server, see "Secure Sockets Layer (SSL)" in *Understanding Security for Oracle WebLogic Server*.

10.2 Validation of Security Certificates

WebLogic Server adds the ability to examine details contained in an end user request to determine whether authentication should succeed or fail. The details may include the end user's certificate, Subject, and IP address. This capability is provided by the `weblogic.security.SSL.CertificateCallback` interface, which you can implement to create a certificate callback handler. When configured with WebLogic Server, this callback handler is invoked automatically whenever a client request is received over a secure RMI connection; for example, one that uses the T3S or IIOPS protocols.

For more information about this feature, see "Checking the Validity of End User Certificates" in *Securing Oracle WebLogic Server*.

10.3 SAML Single Sign-On Attribute Support

In this release, WebLogic Server enhances the SAML 1.1 and 2.0 Credential Mapping provider and Identity Assertion provider mechanisms to support the use of a custom attribute mapper that can obtain additional attributes (other than group information) to be written into SAML assertions, and to then map attributes from incoming SAML assertions.

For more information, including details about the SAML attribute APIs that have been added for this enhancement and examples, see "Configuring SAML SSO Attribute Support" in *Programming Security for Oracle WebLogic Server*.

11 WebLogic Service Component Architecture (WebLogic SCA)

Oracle Web Services Manager (OWSM) policies are now supported in WebLogic SCA applications. For information, see *Developing WebLogic SCA Applications for Oracle WebLogic Server*.

12 Web Applications, Servlets, and JSPs

This section describes changes and new Web application, servlet, and JSP features in this release of WebLogic Server.

- [ActiveCache](#)
- [Classloader Analysis](#)

12.1 ActiveCache

In this release of WebLogic Server, you can use the WebLogic Server Administration Server, via the Administration Console or WLST, and java-based Node Manager to manage and monitor the life cycle of stand-alone Coherence cache servers. For more information, see "Starting Cache Servers Using Node Manager" in *Using ActiveCache*.

12.2 Classloader Analysis

WebLogic Server now provides a Web-based class analysis tool which simplifies filtering classloader configuration and aids you in analyzing classloading issues, such as detecting conflicts, debugging application classpaths and class conflicts, and proposes solutions to help you resolve them. For more information, see "Using the Classloader Analysis Tool (CAT)" in *Developing Applications for Oracle WebLogic Server*.

13 Web Services

This section describes new and changed WebLogic Web Services features in this release of WebLogic Server.

- [Support for WS-I Reliable Secure Profile \(RSP\) for JAX-WS Web Services](#)
- [Certification With JAX-RS Jersey Reference Implementation \(RI\) for Programming RESTful Web Services](#)
- [Enhanced Monitoring Support of WebLogic Java EE Web Services Using Fusion Middleware Control](#)
- [New SAML 1.1 Bearer Token Over HTTPS Example](#)
- [Support for SAML Attributes in Web Services](#)

13.1 Support for WS-I Reliable Secure Profile (RSP) for JAX-WS Web Services

This release provides support for WS-I RSP 1.0 for JAX-WS Web services, including support for the standards defined in [Table 1](#).

For more information about the features supported, see:

- ["Using Web Services Reliable Messaging" in *Programming Advanced Features of JAX-WS Web Services for Oracle WebLogic Server*](#).
- ["Invoking a Web Service Asynchronously" in *Programming Advanced Features of JAX-WS Web Services for Oracle WebLogic Server*](#).
- [Securing WebLogic Web Services for Oracle WebLogic Server](#)

Table 1 *WS-I RSP Standards Supported for JAX-WS Web Services*

Standard	Description
Web Services Reliable Messaging Version 1.2	Implementation that enables two Web services running on different WebLogic Server instances to communicate reliably in the presence of failures in software components, systems, or networks.
Web Services MakeConnection Version 1.1	Provides a mechanism for the transfer of messages between two endpoints when the sending endpoint is unable to initiate a new connection to the receiving endpoint.
Web Services SecurityPolicy 1.3	Set of security policy assertions for use with the WS-Policy framework.
Web Services Trust Language Version 1.4	Extensions that build on Web Services Security (WS-Security) to secure asynchronous communication.
Web Services SecureConversation Version 1.4	Extensions that build on WS-Security and WS-Trust to secure asynchronous communication.

13.2 Certification With JAX-RS Jersey Reference Implementation (RI) for Programming RESTful Web Services

WebLogic Server ships with a set of pre-built shared libraries, packaged as Web applications, that are required to run RESTful Web service applications that are based

on the Jersey JAX-RS RI Version 1.1.5.1. You can register and use this version with WebLogic Server or build and deploy a more recent version, as required. For more information, see "Programming RESTful Web Services" in *Programming Advanced Features of JAX-WS Web Services for Oracle WebLogic Server*.

13.3 Enhanced Monitoring Support of WebLogic Java EE Web Services Using Fusion Middleware Control

In this release, Oracle Enterprise Manager Fusion Middleware Control provides enhanced monitoring support for WebLogic Java EE Web services, specifically policy violation information. For details, see "Viewing the Security Violations for a Web Service" in *Security and Administrator's Guide for Web Services*.

13.4 New SAML 1.1 Bearer Token Over HTTPS Example

This release of WebLogic Server includes a SAML 1.1 Bearer example with a standalone client. This example is available in the WebLogic Server installation in `WL_HOME\samples\server\examples\src\examples\webservices\saml\bearer11ss1`. It demonstrates the minimum configuration and setup with WS-Trust to use a SAML assertion for authentication of a Web service application.

13.5 Support for SAML Attributes in Web Services

This release of WebLogic Server includes support for SAML Attributes in Web services. The SAML specification (see <http://www.oasis-open.org>) allows additional, unspecified information about a particular subject to be exchanged between SAML partners as attribute statements in an assertion. A SAML attribute assertion is therefore a particular type of SAML assertion that conveys site-determined information about attributes of a Subject. Attribute data is of type `String`.

SAML attributes can be examined on the target partner service, and they can be used as extra information for authentication or authorization.

This release includes a sample application that implements SAML attributes for SAML 2.0. This example is available in the WebLogic Server installation in `WLS_HOME\samples\server\examples\src\examples\webservices\saml\saml20sv`.

See "Configuring SAML Attributes in a Web Service" in *Securing WebLogic Web Services for Oracle WebLogic Server* for additional information and examples.

14 Deprecated Functionality (WebLogic Server 11g Release 1)

Information about deprecated functionality for WebLogic Server 11g Release 1 can be found on My Oracle Support at <https://support.oracle.com/>. Enter the following document ID in the **Search Knowledge Base** field:

888028.1

15 Deprecated Functionality (WebLogic Server 10.3)

This section lists all functionality that was deprecated in WebLogic Server 10.3.

- [WebLogic Server Java Utilities](#)
- [Oracle Type 4 JDBC Driver](#)

- [Deployment](#)
- [OpenJPA](#)
- [Apache Beehive Support](#)

15.1 WebLogic Server Java Utilities

The command line tool *EarInit*, documented in the *Command Reference for Oracle WebLogic Server*, has been deprecated in this release of WebLogic Server. As a result, you should no longer:

- Use the **DDInit** utility to generate deployment descriptors for Enterprise applications.
- Use the **ddcreate** ant task, which calls **EarInit**.

15.2 Oracle Type 4 JDBC Driver

The Oracle Type 4 JDBC driver has been deprecated in WebLogic Server.10.3. It has been removed in WebLogic Server 10.3.1. Instead of this driver, you should use the Oracle Thin Driver that is provided with WebLogic Server. For details about the Oracle Thin Driver, see "Using JDBC Drivers with WebLogic Server" in *Configuring and Managing JDBC for Oracle WebLogic Server*.

15.3 Deployment

Internal fields and methods in the following classes have been deprecated in this release of WebLogic Server, and are no longer documented.

- `weblogic.deploy.api.model.WebLogicDeployableObject`
- `weblogic.deploy.api.model.WebLogicJ2eeApplicationObject`
- `weblogic.deploy.api.shared.WebLogicModuleType`
- `weblogic.deploy.api.tools.SessionHelper`

See the following sections for a complete list.

15.3.1 `weblogic.deploy.api.model.WebLogicDeployableObject`

This section lists the deprecated fields, methods, and classes for `weblogic.deploy.api.model.WebLogicDeployableObject`.

Fields

String uri

Boolean haveAppRoot

DDRootFields ddRoot

ClassLoaderControl clf

File Plan

File plandir

DeploymentPlanBean planBean

LibrarySpec[] libraries

boolean deleteOnClose

ClassFinder resourceFinder
InputStream getDDStream()
void setDDBeanRoot()
InputStream getStreamFromParent()

Methods

LibrarySpec[] getLibraries()
WebLogicJ2EEApplicationObject getParent()
void closeGCL()
void closeResourceFinder()
void closeVJF()

Class

DDRootFields

15.3.2 weblogic.deploy.api.model.WebLogicJ2eeApplicationObject

This section lists the deprecated fields and methods for weblogic.deploy.api.model.WebLogicJ2eeApplicationObject.

Fields

ApplicationBean app

Methods

String[] getModuleUris()
void initEmbeddedModules()
void addModule()
File getModulePath

15.3.3 weblogic.deploy.api.shared.WebLogicModuleType

This section lists deprecated fields for weblogic.deploy.api.shared.WebLogicModuleType.

Fields

WebLogicModuleType CONFIG
WebLogicModuleType SUBMODULE
String MODULETYPE_EAR
String MODULETYPE_WAR
String MODULETYPE_EJB
String MODULETYPE_RAR
String MODULETYPE_CAR
String MODULETYPE_UNKNOWN
String MODULETYPE_JMS

String MODULETYPE_JDBC
String MODULETYPE_JDBC
String MODULETYPE_INTERCEPT
String MODULETYPE_CONFIG

15.3.4 **weblogic.deploy.api.tools.SessionHelper**

This section lists deprecated methods for `weblogic.deploy.api.tools.SessionHelper`.

Methods

`void setDebug()`
`SessionHelper()`
`LibrarySpec registerLibrary()`
`LibrarySpec[] getLibraries()`
`void enableLibraryMerge()`
`void bumpVersion()`

15.4 OpenJPA

OpenJPA now has a set of APIs for which compatibility is guaranteed. These are the public interfaces and annotations in the `org.apache.openjpa.persistence` and `org.apache.openjpa.persistence.jdbc` packages. To ensure this compatibility, the return type for some method signatures on these interfaces were changed in non-backward compatible ways (see [Section 15.4.1, "OpenJPA Changed Method Signatures"](#)). Other methods and fields were deprecated in OpenJPA 1.0, making it likely that they will be removed in a future release of OpenJPA (see [Section 15.4.2, "OpenJPA Deprecated Methods and Fields"](#)). Therefore, their use cannot be relied on.

Note: Only the OpenJPA interfaces and classes marked `@published` have compatibility guarantees. The OpenJPA project strives to maintain compatibility for the SPI interfaces, but does not provide any guarantees on them. Additionally, classes and interfaces navigable from the SPI interfaces may change in the future.

15.4.1 OpenJPA Changed Method Signatures

This section lists the OpenJPA changed method signatures.

Table 2 *org.apache.openjpa.persistence.OpenJPAEntityManager Changed Method Signatures*

Pre-1.0 method signature	Method signature for 1.0 and greater
<code>public int getConnectionRetainMode();</code>	<code>public ConnectionRetainMode getConnectionRetainMode();</code>
<code>public int getRestoreState();</code>	<code>public RestoreStateType getRestoreState();</code>

Table 2 (Cont.) *org.apache.openjpa.persistence.OpenJPAEntityManager* Changed Method Signatures

Pre-1.0 method signature	Method signature for 1.0 and greater
public int getDetachState();	public DetachStateType getDetachState();
public int getAutoClear();	public AutoClearType getAutoClear();
public int getAutoDetach();	public EnumSet<AutoDetachType> getAutoDetach();

Table 3 *org.apache.openjpa.persistence.OpenJPAQuery* Changed Method Signatures

Pre-1.0 method signature	Method signature for 1.0 and greater
public int getOperation();	public QueryOperationType getOperation();

Table 4 *org.apache.openjpa.persistence.jdbc.JDBCFetchPlan* Changed Method Signatures

Pre-1.0 method signature	Method signature for 1.0 and greater
public int getEagerFetchMode();	public FetchMode getEagerFetchMode();
public int getSubclassFetchMode();	public FetchMode getSubclassFetchMode();
public int getResultSetType();	public ResultSetType getResultSetType();
public int getFetchDirection();	public FetchDirection getFetchDirection();
public int getJoinSyntax();	public JoinSyntax getJoinSyntax();

Table 5 *org.apache.openjpa.persistence.jdbc.EagerFetchMode*

Pre-1.0 method signature	Method signature for 1.0 and greater
EagerFetchType value() default EagerFetchType.NONE;	FetchMode value() default FetchMode.NONE;

Table 6 *org.apache.openjpa.persistence.jdbc.SubclassFetchMode*

Pre-1.0 method signature	Method signature for 1.0 and greater
EagerFetchType value() default EagerFetchType.NONE;	FetchMode value() default FetchMode.NONE;

15.4.2 OpenJPA Deprecated Methods and Fields

This section lists the OpenJPA deprecated methods and fields.

Table 7 *org.apache.openjpa.persistence*

Deprecated	Use Instead
OpenJPAPersistence.EntityManager	JPAFacadeHelper
OpenJPAPersistence.EntityManagerFactory	JPAFacadeHelper
OpenJPAPersistence.toEntityManagerFactory (BrokerFactory)	JPAFacadeHelper
OpenJPAPersistence.toBrokerFactory(EntityManagerFactory)	JPAFacadeHelper

Table 7 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
OpenJPAPersistence.toEntityManager(Broker)	JPAFacadeHelper
OpenJPAPersistence.toBroker(EntityManager)	JPAFacadeHelper
OpenJPAPersistence.getMetaData(Object)	JPAFacadeHelper
OpenJPAPersistence.getMetaData(EntityManager, Class)	JPAFacadeHelper
OpenJPAPersistence.getMetaData(EntityManagerFactory, Class)	JPAFacadeHelper
OpenJPAPersistence.fromOpenJPAObjectId(Object)	JPAFacadeHelper
OpenJPAPersistence.toOpenJPAObjectId(Class, Metadata, Object)	JPAFacadeHelper
OpenJPAPersistence.toOpenJPAObjectId(Class, Metadata, Object[])	JPAFacadeHelper
OpenJPAPersistence.toOpenJPAObjectId(Class, Metadata, Collection)	JPAFacadeHelper
OpenJPAPersistence.fromOpenJPAObjectIdClasses(Class)	JPAFacadeHelper
FetchPlan.getQueryResultCache()	FetchPlan.getQueryResultCacheEnabled()
FetchPlan.setQueryResultCache(boolean cache)	FetchPlan.setQueryResultCache()
FetchPlan.getDelegate()	FetchPlan.getDelegate() Note: Cast to ExtentImpl. This method pierces the published-API boundary, as does the SPI cast.
OpenJPAEntityManagerFactory.CONN_RETAIN_DEMAND	ConnectionRetainMode enum
OpenJPAEntityManagerFactory.CONN_RETAIN_TRANS	ConnectionRetainMode enum
OpenJPAEntityManagerFactory.CONN_RETAIN_ALWAYS	ConnectionRetainMode enum
OpenJPAEntityManagerFactory.getConfiguration()	OpenJPAEntityManagerFactorySPI.getConfiguration()
OpenJPAEntityManagerFactory.addLifecycleListener(Object, Class[])	OpenJPAEntityManagerFactorySPI.addLifecycleListener(Object, Class[])
OpenJPAEntityManagerFactory.removeLifecycleListener(Object)	OpenJPAEntityManagerFactorySPI.removeLifecycleListener(Object)
OpenJPAEntityManagerFactory.addTransactionListener(Object)	OpenJPAEntityManagerFactorySPI.addTransactionListener(Object)
OpenJPAEntityManagerFactory.removeTransactionListener(Object)	OpenJPAEntityManagerFactorySPI.removeTransactionListener(Object)
QueryResultCache.getDelegate()	QueryResultCache.getDelegate() Note: Cast to ExtentImpl. This method pierces the published-API boundary, as does the SPI cast.

Table 7 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
Extent.getDelegate()	Extent.getDelegate() Note: Cast to ExtentImpl. This method pierces the published-API boundary, as does the SPI cast.
OpenJPAQuery.OP_SELECT	QueryOperationType enum
OpenJPAQuery.OP_DELETE	QueryOperationType enum
OpenJPAQuery.OP_UPDATE	QueryOperationType enum
OpenJPAQuery.FLUSH_TRUE	FlushModeType enum
OpenJPAQuery.FLUSH_FALSE	FlushModeType enum
OpenJPAQuery.FLUSH_WITH_CONNECTIONS	FlushModeType enum
OpenJPAQuery.addFilterListener(FilterListener)	OpenJPAQuerySPI.AddFilterListener(FilterListener)
OpenJPAQuery.removeFilterListener(FilterListener)	OpenJPAQuerySPI.removeFilterListener(FilterListener)
OpenJPAQuery.addAggregateListener(AggregateListener)	OpenJPAQuerySPI.addAggregateListener(AggregateListener)
OpenJPAQuery.removeAggregateListener(AggregateListener)	OpenJPAQuerySPI.removeAggregateListener(AggregateListener)
StoreCache.getDelegate()	StoreCache.getDelegate() Note: Cast to ExtentImpl. This method pierces the published-API boundary, as does the SPI cast.
Generator.getDelegate()	Generator.getDelegate() Note: Cast to ExtentImpl. This method pierces the published-API boundary, as does the SPI cast.
OpenJPAEntityManager.CONN_RETAIN_DEMAND	ConnectionRetainMode enum
OpenJPAEntityManager.CONN_RETAIN_TRANS	ConnectionRetainMode enum
OpenJPAEntityManager.CONN_RETAIN_ALWAYS	ConnectionRetainMode enum
OpenJPAEntityManager.DETACH_FETCH_GROUPS	DetachStateType enum
OpenJPAEntityManager.DETACH_FGS	DetachStateType enum
OpenJPAEntityManager.DETACH_LOADED	DetachStateType enum
OpenJPAEntityManager.DETACH_ALL	DetachStateType enum
OpenJPAEntityManager.RESTORE_ALL	RestoreStateType enum
OpenJPAEntityManager.RESTORE_NONE	RestoreStateType enum
OpenJPAEntityManager.RESTORE_IMMUTABLE	RestoreStateType enum
OpenJPAEntityManager.DETACH_CLOSE	AutoDetachType enum

Table 7 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
OpenJPAEntityManager.DETACH_COMMIT	AutoDetachType enum
OpenJPAEntityManager.DETACH_NONTXREAD	AutoDetachType enum
OpenJPAEntityManager.DETACH_ROLLBACK	AutoDetachType enum
OpenJPAEntityManager.CLEAR_DATASTORE	AutoClearType enum
OpenJPAEntityManager.CLEAR_ALL	AutoClearType enum
OpenJPAEntityManager.CALLBACK_FAIL_FAST	CallBackMode enum
OpenJPAEntityManager.CALLBACK_IGNORE	CallBackMode enum
OpenJPAEntityManager.CALLBACK_LOG	CallBackMode enum
OpenJPAEntityManager.CALLBACK_RETHROW	CallBackMode enum
OpenJPAEntityManager.CALLBACK_ROLLBACK	CallBackMode enum
OpenJPAEntityManager.getConfiguration()	OpenJPAEntityManagerSPI.getConfiguration()
OpenJPAEntityManager.setRestoreState(int)	OpenJPAEntityManager.setRestoreState(RestoreStateType)
OpenJPAEntityManager.setDetachState(int)	OpenJPAEntityManager.setDetachState(DetachStateType)
OpenJPAEntityManager.setAutoClear(int)	OpenJPAEntityManager.setAutoClear(AutoClearType)
OpenJPAEntityManager.setAutoDetach(int)	OpenJPAEntityManager.setAutoDetach(AutoDetachType)
OpenJPAEntityManager.setAutoDetach(int, boolean)	OpenJPAEntityManager.setAutoDetach(AutoDetachType, boolean)
OpenJPAEntityManager.isLargeTransaction()	OpenJPAEntityManager.isTrackChangesByType()
OpenJPAEntityManager.setLargeTransaction(boolean)	OpenJPAEntityManager.setTrackChangesByType(boolean)
OpenJPAEntityManager.addTransactionListener(Object)	OpenJPAEntityManagerSPI.addTransactionListener(Object)
OpenJPAEntityManager.removeTransactionListener(Object)	OpenJPAEntityManagerSPI.removeTransactionListener(Object)
OpenJPAEntityManager.getTransactionListenerCallbackMode()	OpenJPAEntityManagerSPI.getTransactionListenerCallbackMode()
OpenJPAEntityManager.setTransactionListenerCallbackMode(int)	OpenJPAEntityManagerSPI.setTransactionListenerCallbackMode(int)
OpenJPAEntityManager.addLifecycleListener(Object, Class[])	OpenJPAEntityManagerSPI.addLifecycleListener(Object, Class[])
OpenJPAEntityManager.removeLifecycleListener(Object)	OpenJPAEntityManagerSPI.removeLifecycleListener(Object)

Table 7 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
OpenJPAEntityManager.getLifecycleListenerCallbackMode()	OpenJPAEntityManagerSPI.getLifecycleListenerCallbackMode()
OpenJPAEntityManager.setLifecycleListenerCallbackMode(int)	OpenJPAEntityManagerSPI.setLifecycleListenerCallbackMode(int)
OpenJPAEntityManager.begin()	EntityTransaction.begin()
OpenJPAEntityManager.commit()	EntityTransaction.commit()
OpenJPAEntityManager.rollback()	EntityTransaction.rollback()
OpenJPAEntityManager.isActive()	EntityTransaction.isActive()
OpenJPAEntityManager.commitAndResume()	OpenJPAEntityTransaction.commitAndResume()
OpenJPAEntityManager.rollbackAndResume()	OpenJPAEntityTransaction.rollbackAndResume()
OpenJPAEntityManager.setRollbackOnly()	EntityTransaction.setRollbackOnly()
OpenJPAEntityManager.setRollbackOnly(Throwable)	OpenJPAEntityTransaction.setRollbackOnly()
OpenJPAEntityManager.getRollbackCause()	OpenJPAEntityTransaction.getRollbackCause()
OpenJPAEntityManager.getRollbackOnly()	EntityTransaction.getRollbackOnly()
JDBCFetchPlan.EAGER_MODE	FetchMode enum
JDBCFetchPlan.EAGER_JOIN	FetchMode enum
JDBCFetchPlan.EAGER_PARALLEL	FetchMode enum
JDBCFetchPlan.SIZE_UNKNOWN	LRSSizeAlgorithm enum
JDBCFetchPlan.SIZE_LAST	LRSSizeAlgorithm enum
JDBCFetchPlan.SIZE_QUERY	LRSSizeAlgorithm enum
JDBCFetchPlan.SYNTAX_SQL92	JoinSyntax enum
JDBCFetchPlan.SYNTAX_TRADITIONAL	JoinSyntax enum
JDBCFetchPlan.SYNTAX_DATABASE	JoinSyntax enum
JDBCFetchPlan.setEagerFetchMode(int)	JDBCFetchPlan.setEagerFetchMode(FetchMode)
JDBCFetchPlan.setSubclassFetchMode(int)	JDBCFetchPlan.setSubclassFetchMode(FetchMode)
JDBCFetchPlan.setResultSetType(int)	JDBCFetchPlan.setResultSetType(ResultSetType)
JDBCFetchPlan.setFetchDirection(int)	JDBCFetchPlan.setFetchDirection(FetchDirection)
JDBCFetchPlan.getLRSSize()	JDBCFetchPlan.getLRSSizeAlgorithm()
JDBCFetchPlan.setLRSSize(int)	JDBCFetchPlan.setLRSSizeAlgorithm(LRSSizeAlgorithm)
JDBCFetchPlan.setJoinSyntax(int)	JDBCFetchPlan.setJoinSyntax(setJoinSyntax)

15.4.3 OpenJPAEntityManager

In WebLogic Server 10g Release 3 (10.3), the `org.apache.openjpa.persistence.OpenJPAEntityManager` interface extends `EntityTransaction`. This relationship is deprecated; in future releases, `OpenJPAEntityManager` will not extend `EntityTransaction`.

The following provides an example of how this might impact your code:

Pre-10.3

```
OpenJPAEntityManager em = ...
EntityTransaction t = em;
```

10.3

```
OpenJPAEntityManager em = ...;
EntityTransaction t = em.getTransaction();
```

15.5 Apache Beehive Support

Apache Beehive has been deprecated as of WebLogic Server 10.3. Oracle intends to remove Apache Beehive APIs in a future WebLogic Server Version release. In preparation, we recommend that you migrate your Beehive applications and infrastructure to other frameworks such as Oracle's ADF or Java Server Faces at your earliest convenience. Note, Beehive will still be available and supported for use within WebLogic Integration and WebLogic Portal.

16 Standards Support

This release of WebLogic Server supports the following standards and versions.

16.1 Java Standards

[Table 8](#) lists currently supported Java standards.

Table 8 Java Standards Support

Standard	Version
JAAS	1.0 Full
Java API for XML-Based Web Services (JAX-WS)	2.1, 2.0
Java Authorization Contract for Containers (JACC)	1.1
Java EE	5.0
Java EE Application Deployment	1.2
Java EE CA	1.5, 1.0
Java EE EJB	3.0, 2.1, 2.0, and 1.1
Java EE Enterprise Web Services	1.2, 1.1
Java EE JDBC	4.0, 3.0
Java EE JMS	1.1, 1.0.2b
Java EE JNDI	1.2
Java EE JSF	2.0, 1.2, 1.1
Java EE JSP	2.1, 2.0, 1.2, and 1.1

Table 8 (Cont.) Java Standards Support

Standard	Version
Java EE Servlet	2.5, 2.4, 2.3, and 2.2
Java RMI	1.0
JavaMail	1.4
JAX-B	2.1, 2.0
JAX-P	1.2, 1.1
JAX-R	1.0
JAX-RPC	1.1, 1.0 (deprecated)
JCE	1.4
JDKs	6.0 (aka 1.6), 5.0 (aka 1.5, clients only)
JMX	1.2, 1.0
JPA	1.0, 2.0 For information on JPA 2.0 support, see Section 4.3, "Support for JPA 2.0 Providers."
JSR 77: Java EE Management	1.1
JSTL	1.2
OTS/JTA	1.2 and 1.1
RMI/IIOP	1.0
SOAP Attachments for Java (SAAJ)	1.3, 1.2
Streaming API for XML (StAX)	1.0
Web Services Metadata for the Java Platform	2.0, 1.1

16.2 Web Services Standards

For the current list of standards supported for WebLogic Web services, see "Features and Standards Supported by WebLogic Web Services" in *Introducing WebLogic Web Services for Oracle WebLogic Server*.

16.3 Other Standards

[Table 9](#) lists other standards that are supported in this release of WebLogic Server.

Table 9 Other Standards

Standard	Version
SSL	v3
X.509	v3
LDAP	v3
TLS	v1
HTTP	1.1
SNMP	SNMPv1, SNMPv2, SNMPv3

Table 9 (Cont.) Other Standards

Standard	Version
xTensible Access Control Markup Language (XACML)	2.0
Partial implementation of Core and Hierarchical Role Based Access Control (RABC) Profile of XACML	2.0
Internet Protocol (IP)	Versions: <ul style="list-style-type: none">▪ v6▪ v4

For more information about IPv6 support for all Fusion Middleware products, refer to the IPv6 Certification worksheet in the *Oracle Fusion Middleware 11g Release 1 (11.1.1.x) Certification Matrix* at

[http://www.oracle.com/technology/software/products/ias/files/oracle%20fusion%20middleware%2011gR1%20\(11.1.1.x\)%20certification%20matrix.xls](http://www.oracle.com/technology/software/products/ias/files/oracle%20fusion%20middleware%2011gR1%20(11.1.1.x)%20certification%20matrix.xls).

17 Supported Configurations

For the most current information on supported configurations, refer to the Oracle Fusion Middleware Supported Configurations Central Hub at

http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html.

18 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit

<http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Oracle Fusion Middleware What's New in Oracle WebLogic Server, 11g Release 1 (10.3.4)
E13852-05

Copyright © 2007, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.