

Oracle® Fusion Middleware

What's New in Oracle WebLogic Server

11g Release 1 (10.3.1)

E13852-01

May 2009

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 version number of Oracle WebLogic Server is 10.3.1.

- [Section 1, "Administration Console"](#)
- [Section 2, "Diagnostics"](#)
- [Section 3, "Enterprise Java Beans \(EJBs\)"](#)
- [Section 4, "Installer"](#)
- [Section 5, "Integration with Oracle WebLogic Suite"](#)
- [Section 6, "JDBC"](#)
- [Section 7, "JTA"](#)
- [Section 8, "JMX"](#)
- [Section 9, "Messaging"](#)
- [Section 10, "Miscellaneous"](#)
- [Section 11, "Security"](#)
- [Section 12, "Spring"](#)
- [Section 13, "Web Services"](#)
- [Section 14, "WLST"](#)
- [Section 15, "Deprecated Functionality \(WebLogic Server 10.3.1\)"](#)
- [Section 16, "Deprecated Functionality \(WebLogic Server 10.3\)"](#)
- [Section 17, "Standards Support"](#)
- [Section 18, "Supported Configurations"](#)
- [Section 19, "Documentation Accessibility"](#)

1 Administration Console

In this release of WebLogic Server:

- A sample Look and Feel is provided, which you can modify to create a custom Look and Feel for the Administration Console.
- Online help can be created and associated with console extensions.

For more information, see *Oracle Fusion Middleware Extending the Administration Console for Oracle WebLogic Server*.

2 Diagnostics

This section describes new features for the WebLogic Server Diagnostics Framework (WLDF).

2.1 New Diagnostic Monitors

Two diagnostic monitors have been added:

- JDBC_After_Reserve_Connection_Internal
- JDBC_After_Release_Connection_Internal

These diagnostic instrumentation monitors can be configured in a WLDF module at the server level. They provide additional visibility when JDBC connections are reserved and released. For more information, see "Diagnostic Monitor Library" in *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

3 Enterprise Java Beans (EJBs)

In this release, you can leverage the features of Oracle TopLink with your EJB 3.0 applications. Oracle TopLink, which is included with the WebLogic Server installation, is an advanced, object-persistence and object-transformation framework that provides development tools and run-time capabilities that reduce development and maintenance efforts, and increase enterprise application functionality. For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle TopLink*.

Note: Oracle Kodo is deprecated in this release.

4 Installer

This section describes changes and new features for the WebLogic Server installer.

4.1 Code Examples and MedRec Sample Application

In this release of WebLogic Server, the WebLogic Server Code Examples and MedRec sample application are not installed by default. To run the code examples and MedRec, you must select a custom installation of WebLogic Server and select to install the Server Examples.

4.2 Obtaining Security Updates from My Oracle Support

When you run the installer, you are now prompted to indicate whether you want to register your WebLogic Server installation with My Oracle Support. By registering, using your My Oracle Support user name and password, Oracle Support will notify

you immediately of any security updates that are specific to your installation. Registration is optional but strongly recommended.

4.3 Oracle Enterprise Pack for Eclipse

For Windows and Linux platforms, you may now download a WebLogic Server net or package installer that includes Oracle Enterprise Pack for Eclipse (OEPE). Using OEPE, you can develop, deploy and debug Oracle WebLogic Server applications. Support for WebLogic Server development includes:

- Full support for WebLogic Server version 10.3.1
- Faster development with virtual EAR technology
- Remote deployment support
- Deployment descriptor editing using a graphical design view
- WebLogic Shared Library support
- XMLBeans support
- EJBGen

For more information about OEPE, see the Oracle Enterprise Pack for Eclipse Documentation, available at <http://www.oracle.com/technology/products/enterprise-pack-for-eclipse/index.html>.

4.4 Middleware Home Directory

Several WebLogic Server scripts now contain a new environment variable to represent the parent directory of the location in which the WebLogic Server product software is installed. This environment variable, `MW_HOME`, is set to this parent directory, which is referred to as the Middleware Home directory.

By default, the Middleware Home directory is set to `Oracle/Middleware`. However, if the installer detects an existing WebLogic installation on your machine, WebLogic Server is installed by default in that location instead, and the `MW_HOME` variable is set to that location accordingly.

The `BEA_HOME` environment variable is still in use in several WebLogic Server scripts, but the documentation now generally refers to the Middleware Home directory instead of BEA Home.

5 Integration with Oracle WebLogic Suite

WebLogic Server now provides the core application server run time within the integrated Oracle WebLogic Suite Java infrastructure.

WebLogic Suite contains the following server-side components:

- Oracle WebLogic Server
- Oracle JRockit
- Oracle Coherence
- Oracle TopLink

This integrated infrastructure enhances application performance, improves application availability, and enables predictable and reliable application scalability with high

quality of service. WebLogic Suite includes highly productive development tools based on Oracle JDeveloper and Oracle Enterprise pack for Eclipse, and a fully integrated management for large-scale administration and operations with the Oracle Enterprise Manager. Taken together, the development, run-time and management capabilities of WebLogic Suite provide the foundation for implementing mission-critical enterprise applications.

6 JDBC

This section describes new JDBC features in this release of WebLogic Server.

6.1 Support for Connecting Data Sources to Oracle RAC Services

Support for connecting data sources and multi data sources to services running on Oracle RAC nodes has been added. To support this feature, two new JDBC driver configuration options have been added to WebLogic Server:

- Oracle's Driver (Thin) for RAC Service-Instance connections
- Oracle's Driver (Thin XA) for RAC Service-Instance connections

For more information, see "Configuring Connections to Services on RAC Nodes" in *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*

6.2 resourcepool.max_test_wait_secs

Use this flag to manage failures, such as the DBMS network cable being severed, which can cause connection tests and applications to hang for very long periods of time. See "Managing WebLogic JDBC Resources" in *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*.

6.3 Sybase JDBC Drivers

The following Sybase JDBC drivers are no longer included with WebLogic Server:

- Sybase jConnect 4.5 (jConnect.jar)
- Sybase jConnect 5.5 (jconn2.jar)
- Sybase jConnect 6.0 (jconn3.jar)

You can download the latest version of the Sybase drivers directly from Sybase.

If you upgraded to WebLogic Server 10.3.1 from a previous version, these Sybase drivers will still be present in the WL_HOME\server\lib folder.

7 JTA

This release changes the behavior of the resource registration name for XA data source configurations. In previous releases, the JTA registration name was simply the name of the data source. Now, the registration name is a combination of data source name and domain.

All resources registered with JTA have a corresponding runtime MBean that exposes XA usage statistics for the resource. This change alters the JMX ObjectName of the MBean, and may impact existing applications that perform a JMX lookup of such a runtime MBean by name. For example, an existing data source configuration with a name of "mydatasource" in domain "mydomain" would have a JTA resource runtime

MBean registered under the object name

```
"com.bea:ServerRuntime=myserver,Name=mydatasource,Type=TransactionResourceRuntime,JTARuntime=JTARuntime." The new object name would be "com.bea:ServerRuntime=myserver,Name=mydatasource_mydomain,Type=TransactionResourceRuntime,JTARuntime=JTARuntime."
```

See "Registering an XAResource to Participate in Transactions" in *Oracle Fusion Middleware Programming JTA for Oracle WebLogic Server*.

8 JMX

This section describes new JMX features in this release of WebLogic Server.

8.1 Support for Registering Custom MBeans in the Domain Runtime MBean Server

This release of WebLogic Server adds the ability to register custom MBeans in the Domain Runtime MBean Server. See "Make Local Connections to the Domain Runtime MBean Server" in *Oracle Fusion Middleware Developing Custom Management Utilities With JMX for Oracle WebLogic Server* and "Registering Custom MBeans in the Domain Runtime Server" in *Oracle Fusion Middleware Developing Manageable Applications With JMX for Oracle WebLogic Server* for more information.

9 Messaging

This release includes support for inter-operating with Oracle Advanced Queuing (AQ) through Foreign JMS and JDBC data source configuration in a WebLogic Server domain. Both local and remote JMS clients can use Oracle AQ destinations from WebLogic JNDI. See "Using AQ JMS" in *Oracle Fusion Middleware Programming JMS for Oracle WebLogic Server*.

10 Miscellaneous

This section describes miscellaneous changes and new features in this release of WebLogic Server.

10.1 Schema Namespaces and Locations Have Been Updated

Namespace URIs and schema locations that contained `www.bea.com` have been changed to refer to `xmlns.oracle.com`. In addition, the WebLogic Server version numbers (920, 90) have been removed.

See the Oracle WebLogic Server Schema home at

<http://www.oracle.com/technology/weblogic/index.html>. Also see, "XML Deployment Descriptors" in *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*.

10.2 HTTP Load Balancing and State Replication

In this release of WebLogic Server, Oracle HTTP Server is supported as a front end Web Server to WebLogic Server configurations. HTTP requests received by Oracle HTTP Server can be routed to WebLogic Server managed servers with load balancing

and session state replication for failover and high availability. See *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server*.

This release of WebLogic Server supports integration of Coherence*Web for HTTP Session State Replication using the Coherence*Web SPI for WebLogic Server. See the Oracle Coherence*Web documentation.

11 Security

This section describes new JMX features in this release of WebLogic Server.

11.1 New LDAP Authentication Providers

Two new LDAP authentication providers have been added to WebLogic Server—the Oracle Internet Directory Authentication Provider and the Oracle Virtual Directory Authentication Provider. These authentication providers can store users and groups in, and read users and groups from, the Oracle Internet Directory and Oracle Virtual Directory LDAP servers, respectively.

For information about configuring and using these new security providers, see "Configuring LDAP Authentication Providers" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

11.2 Default Administrator Password Has Changed

The default administrator password for WebLogic Server has been changed from `weblogic` to `welcome1`. Use `weblogic/welcome1` for logging in to the Administration Console when accessing the Examples Server (`wl_server`) domain and Medical Records Server (`medrec`) domain.

12 Spring

In this release of WebLogic Server, WebLogic Spring support has been enhanced as follows:

- Configuration is simplified. Spring applications deployed to WebLogic Server automatically have available to them for dependency injection a number of preconfigured beans. No declaration is required in Spring configuration files.
- The Spring console extension is included as part of WebLogic Server and has only to be enabled.
- The Spring console extension has new monitoring pages that display runtime statistics for Spring applications.
- A new sample application, MedRec (Spring) demonstrates Spring application development practices.

For more information, see *Oracle Fusion Middleware Spring Support in Oracle WebLogic Server*

13 Web Services

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

13.1 New Development Tools

You can develop WebLogic Web services using the following tools:

- **Oracle JDeveloper**—Provides powerful tools that help you discover and use existing Web services, and develop and deploy new Web services. For more information, see "Developing with Web Services" in the "Designing and Developing Applications" section of the Oracle JDeveloper online help.
- **Oracle Enterprise Pack for Eclipse (OEPE)**—Provides a collection of plug-ins to the Eclipse IDE platform that facilitate development of WebLogic Web services. For more information, see the Eclipse IDE platform online help.

13.2 Integration with Oracle Enterprise Manager Fusion Middleware Control

Using Oracle Enterprise Manager Fusion Middleware Control, you can test and monitor Oracle WebLogic Web services. For more information, refer to the following sections in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*:

- "Testing Web Services"
- "Monitoring the Performance of Web Services"

13.3 Support for Oracle WSM Security Policies

You can attach a subset of the Oracle WebLogic Services Manager (WSM) security policies. For complete details, see "Using Oracle Web Service Security Policies" in *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server*.

13.4 Web Service Security Enhancements

This release supports the following security enhancements:

- WS-SecureConversation 1.3 on JAX-WS. For more information, see "Configuring and Using Security Contexts and Derived Keys (WS-SecureConversation)" in *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server*.
- MTOM with WS-Security on JAX-WS.

14 WLST

The section describes new and changed WLST features in this release of WebLogic Server.

14.1 Support for Using WLST With Oracle Fusion Middleware Products

Support for using WLST with some Oracle Fusion Middleware products. Custom WLST commands for these products are described in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*. These commands are WLST versions of many of the Oracle ASCTL commands for Fusion Middleware products.

14.2 Ability to Access and Perform Operations on Custom MBeans Registered in the Domain Runtime MBean Server

A new `domainCustom()` command that lets you access and perform operations on custom MBeans that have been registered in the Domain Runtime MBean Server. See "Accessing Custom MBeans on the Domain Runtime Server" in *Oracle Fusion*

Middleware Oracle WebLogic Scripting Tool, and "domainCustom" in Oracle Fusion Middleware WebLogic Scripting Tool Command Reference.

14.3 Support for Adding Command Group Help and Command Help for Custom WLST Commands

Two new WLST commands, `addHelpCommandGroup()` and `addHelpCommand()`, which lets you add command group help and command help for custom WLST commands to the WLST integrated help. See "addHelpCommandGroup" and "addHelpCommand" in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

14.4 Jython Version

As of this release of WebLogic Server, WLST is based on Jython 2.2.1.

15 Deprecated Functionality (WebLogic Server 10.3.1)

Information about deprecated functionality for WebLogic Server 10.3.1 can be found on My Oracle Support at <https://metalink.oracle.com>.

16 Deprecated Functionality (WebLogic Server 10.3)

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

- [Section 16.1, "WebLogic Server Java Utilities"](#)
- [Section 16.2, "Oracle Type 4 JDBC Driver"](#)
- [Section 16.3, "Deployment"](#)
- [Section 16.4, "OpenJPA"](#)
- [Section 16.5, "Apache Beehive Support"](#)

16.1 WebLogic Server Java Utilities

The command line tool *EarInit*, documented in the *Oracle Fusion Middleware 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**.

16.2 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 *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*.

16.3 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.

16.3.1 `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 getSteamFromParent()

Methods

LibrarySpec[] getLibraries()

WebLogicJ2EEApplicationObject getParent()

void closeGCL()

void closeResourceFinder()

void closeVJF()

Class

DDRootFields

16.3.2 `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

16.3.3 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

16.3.4 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()

16.4 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 16.4.1, "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 16.4.2, "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.

16.4.1 OpenJPA Changed Method Signatures

This section lists the OpenJPA changed method signatures.

Table 1 *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>
<code>public int getDetachState();</code>	<code>public DetachStateType getDetachState();</code>
<code>public int getAutoClear();</code>	<code>public AutoClearType getAutoClear();</code>
<code>public int getAutoDetach();</code>	<code>public EnumSet<AutoDetachType> getAutoDetach();</code>

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

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

Table 3 *org.apache.openjpa.persistence.jdbc.JDBCFetchPlan Changed Method Signatures*

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

Table 4 *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 5 *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;

16.4.2 OpenJPA Deprecated Methods and Fields

This section lists the OpenJPA deprecated methods and fields.

Table 6 *org.apache.openjpa.persistence*

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

Table 6 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
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.
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.

Table 6 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
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
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)

Table 6 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
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)
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

Table 6 (Cont.) org.apache.openjpa.persistence

Deprecated	Use Instead
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)

16.4.3 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;
```

16.5 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.

17 Standards Support

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

17.1 Java Standards

[Table 7](#) lists currently supported Java standards.

Table 7 Java Standards Support

Standard	Version
Java EE	5.0
JDKs	6.0 (aka 1.6), 5.0 (aka 1.5, clients only)
Java EE Enterprise Web Services	1.2, 1.1
Web Services Metadata for the Java Platform	2.0, 1.1
Java API for XML-Based Web Services (JAX-WS)	2.1, 2.0
Java EE EJB	3.0, 2.1, 2.0, and 1.1
Java EE JMS	1.1, 1.0.2b
Java EE JDBC	4.0, 3.0
Java EE JNDI	1.2
OTS/JTA	1.2 and 1.1
Java EE Servlet	2.5, 2.4, 2.3, and 2.2
Java EE Application Deployment	1.2
Java Authorization Contract for Containers (JACC)	1.1
Java EE JSP	2.1, 2.0, 1.2, and 1.1
JSTL	1.2
RMI/IIOP	1.0
JMX	1.2, 1.0
JavaMail	1.2
JAAS	1.0 Full
Java EE CA	1.5, 1.0
Java EE JSF	1.2, 1.1
JCE	1.4
Java RMI	1.0
JAX-B	2.1, 2.0
JAX-P	1.2, 1.1
JAX-RPC	1.1, 1.0 (deprecated)
JAX-R	1.0
SOAP Attachments for Java (SAAJ)	1.3, 1.2
Streaming API for XML (StAX)	1.0
JSR 77: Java EE Management	1.1

17.2 Web Services Standards

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

Table 8 Web Services Standards Support

Standard	Version
Web Services Java EE	1.2, 1.1
Web Services Metadata for the Java Platform (JWS)	2.0, 1.0
Java API for XML-Based Web Services (JAX-WS)	2.1, 2.0
Simple Object Access Protocol (SOAP)	1.1, 1.2
Web Services Description Language (WSDL)	1.1
Java API for XML-based RPC (JAX-RPC)	1.1, 1.0 (deprecated)
SOAP with Attachments for Java (SAAJ)	1.3, 1.2
Web Services Security (WS-Security)	1.1, 1.0
Web Services Policy Framework (WS-Policy)	1.5, 1.2
Web Services Security Policy (WS-SecurityPolicy)	1.2
Web Services Policy Attachment (WS-PolicyAttachment)	1.5, 1.2
Web Services Addressing (WS-Addressing)	1.0, 2004/2008 member submission
Web Services Reliable Messaging (WS-ReliableMessaging)	1.1, 1.0
Web Services Trust Language (WS-Trust)	1.3
Web Services Secure Conversation Language (WS-SecureConversation)	1.3
Universal Description, Discovery, and Integration (UDDI)	2.0
Java API for XML Registries (JAX-R)	1.0
Java Architecture for XML Binding (JAX-B)	2.1, 2.0
Security Assertion Markup Language (SAML)	2.0, 1.1
SAML Token Profile	1.1, 1.0

17.3 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

18 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.

19 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.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

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.