ENTERPRISE JAVA BEANS SUPPORT IN ORACLE APPLICATION SERVER 10G 10.1.3.1 FEATURE OVERVIEW

KEY FEATURES

- EJB 2.1
- TopLink CMP Integration
- Support for Enterprise JavaBeans (EJB) 3.0
- EJB 3.0 Java Persistence API
- EJB 3.0 Web services with web services metadata
- Support for third-party JMS providers
- New EJB state replication policies
- Improved deployment performance for large EJB applications

Oracle Application Server 10g, a component of Oracle Fusion Middleware, represents the next generation of enterprise application server. With its service oriented architecture foundation and grid enabled deployment infrastructure, Oracle Application Server a leader in the Enterprise Java Beans technology provides support for latest specifications of Enterprise Java Beans and facilitates development and deployment robust, performant and scalable business logic and persistence layer of service-oriented application.

Overview

Oracle Application Server 10g (10.1.3.1) being a J2EE 1.4 compliant application server has support for EJB 2.1. It also includes full support for EJB 3.0 and JPA. Oracle is a co-specification lead of the EJB 3.0 specification.

EJB 2.1 Features

The EJB container in OC4J 10.1.3.1 is fully compliant with the EJB 2.1 specification. This includes support for timers, stateless EJB Web services and EJB-QL extensions.

Timer

EJB 2.1 introduced the concept of a timer service, which enables developers to create a program that can schedule a business process to occur at a predetermined time or at a regular interval. The EJB container manages the timer service to allow EJB methods to register to call back at a scheduled time or regular interval. Timers are persistent, and they survive container crashes and shutdown. Transactions are important for enterprise applications, and, for good reasons, timers are transactional. The creation and cancellation of a timer are done within the scope of a transaction and conform to ACID properties. If the transaction is rolled back, the creation or cancellation of the timer is undone.

Oracle Application Server 10g (10.1.3.1) supports timers in all types of EJBs namely Session Beans, Message Driven Beans and Entity beans.

Oracle Application Server 10g (10.1.3.1) also supports a proprietary extension to support cron based timers with EJB.
EJB QL enhancements
OC4J 10.1.3 being compliant to EJB 2.1 has support for several extensions introduced as part of EJB 2.1. It includes support for ORDER BY clause and group functions such as MAX, MIN, AVG, SUM and COUNT in EJB SQL queries.

MDB with JCA Resource Adapter
EJB 2.1 enhanced Message Driven Beans to support messaging providers using Java Connector Architecture and there are two primary benefits to this:

- It allows MDB to be used with non-JMS providers
- It will allows JMS providers with a to be integrated in a standardized manner.

OC4J fully supports messaging providers such as Oracle Enterprise Messaging Services, IBM Websphere MQ, Tibco Enterprise JMS and Sonic MQ.

Stateless EJB as web service
OC4J supported exposing a stateless EJB as a web service since OC4J 9.0.2, however the configuration was Oracle specific. EJB 2.1 standardized exposing Stateless EJB as a web service and provides tools and utilities to deploy an existing EJB as a web service.

Oracle TopLink CMP Persistence Provider
Oracle TopLink is a robust, performant and scalable Object-Relational Mapping framework that can be used with regular Java objects (POJOs), BMP and CMP entity beans. In OC4J 10.1.3.1, Oracle TopLink replaces the old persistence provider aka OrionCMP as the default persistence provider for CMP entity beans. Existing J2EE applications with CMP entity beans using Orion CMP (O-R mapping defined in orion-ejb-jar.xml) will continue to work in OC4J 10.1.3.1, however customers are recommended to migrate their applications to use TopLink CMP provider. Oracle provided a migration utility to migrate applications using Orion CMP to use TopLink CMP provider.

Third-party JMS Provider Support
OC4J 10.1.3.1 has integrated third-party JMS providers such as IBM Websphere MQ, Tibco Enterprise JMS and Sonic. These JMS-providers can be used with MDB using the generic JMS connector shipped with OC4J and participate in distributed transactions.

EJB Lookup Improvement
There are several enhancements in EJB lookup in OC4J 10.1.3.1 and these includes

- Ability to lookup EJBs deployed in remote OC4J without having to hardcode the JNDI properties and ability to declaratively reference a remote EJB.
- Ability to lookup an EJB in another application without having to use RMIInitialContextFactory
- Ability to lookup an EJB without specifying the application name in the
provider URL

- Anonymous lookup of EJB i.e. ability to lookup an EJB without specifying principal and credential. Oracle does not recommend anonymous lookup of EJB for security reasons.
- Ability to lookup a local EJB without having to specify ejb-local-ref in the deployment descriptor.

**BMP Caching Improvement**
OC4J 10.1.3.1 adds the support for commit option A for BMP entity beans and Read-only BMP entity beans. These options improve performance of BMP entity beans by reducing callback methods and thus reducing database operations.

**EJB Deployment Improvement**
OC4J implemented dynamic proxy for EJB wrappers in OC4J 10.1.3.1 and hence there is significant improvement in performance of deployment of large EJB applications. If you have several small ejb-jar modules in the same enterprise application you can set the compilation mode to either batch mode.

**New EJB State Replication Capabilities**
OC4J provides a flexible framework for creating a clustered application environment for Stateful session beans. The OC4J clustering framework supports:

- Replication of objects and values contained in a stateful session EJB instance.
- In-memory replication using multicast or peer-to-peer communication, or persistence of state data to a database.
- Load balancing of incoming requests across OC4J instances.
- Transparent failover across applications within the cluster.
- Configuration within an OC4J instance at either the global or the application level.

Application-level clustering can be configured at the time the application is deployed, or can be enabled by default for all applications that support replication within the OC4J instance.

**Security Enhancements**
OC4J 10.1.3.1 provides a number of enhancements to the base security infrastructure that can be used with EJB applications. These include:

- Support of JAAS mode in the EJB container
- Security context propagation, also known as subject propagation, between two OC4J instances is now supported. Security Context propagation requires JAAS mode to be enabled for both OC4J containers.

In addition, OC4J now supports ORMI over SSL (ORMIS), which enables Secure Socket Layer (SSL) RMI communication between EJB container and remote clients.
The following table summarizes the features included in EJB 2.1 and general enhancements in the EJB container.

<table>
<thead>
<tr>
<th>EJB 2.1</th>
<th>Misc. Improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Timer</td>
<td>• TopLink CMP Integration</td>
</tr>
<tr>
<td>• Connector MDB</td>
<td>• BMP Caching</td>
</tr>
<tr>
<td>• EJBQL Enhancement</td>
<td>• Third-party JMS provider support</td>
</tr>
<tr>
<td>• Stateless EJB as web service</td>
<td>• Transaction Context Propagation</td>
</tr>
<tr>
<td>Security Improvements</td>
<td>• Dynamic proxy generation and EJB Deployment performance</td>
</tr>
<tr>
<td>• Security Context Propagation and JAAS Mode</td>
<td>• EJB Lookup improvements</td>
</tr>
<tr>
<td></td>
<td>• ORMI/SSL</td>
</tr>
</tbody>
</table>

**EJB 3.0**

The goal of **Enterprise JavaBeans (EJB) 3.0** is to simplify development of Java applications and standardize the persistence API for the Java platform. EJB 3.0 is part of the J2EE platform, J2EE 5.0. EJB 3.0 was worked under Java Specification Request 220 (JSR-220) in the Java Community Process (JCP) and Oracle is a co-lead of the specification and produced the Reference Implementation of EJB 3.0 Java Persistence API.

OC4J 10.1.3.1 has full support for EJB 3.0 and JPA.

**Simplified EJB**

EJB 3.0 eliminates the need for home and component interfaces and the requirement for bean classes for implementing `javax.ejb.EnterpriseBean` interfaces. The EJB bean class can be a pure Java class often referred as POJO and the interface can be a simple business interface (POJI). OC4J 10.1.3.1 fully supports simplified programming model for all EJB types.

**Use of Annotations instead of Deployment Descriptors**

Metadata annotation is being used as an alternative to deployment descriptors. Annotations can be used to specify bean types, different attributes such as transaction or security settings and injection of environment or resource references.

Metadata Annotations used for resource references, security setting and common lifecycle methods have been defined in the Common Metadata Annotations for the Java Platform (JSR-250). OC4J 10.1.3.1 has full support for common metadata annotations in the EJB container.

Deployment descriptor can be used to override metadata annotations.
Dependency Injection
The API for lookup and usage of EJB environment and resource references is being simplified and dependency injection is being used instead. Metadata annotation is being used for dependency injection. In OC4J 10.1.3.1, you can optionally use deployment descriptor to define injection of resources.

Enhanced Lifecycle Methods and Callback Listener Classes
Unlike previous versions of EJB, the bean developers do not have implement all unnecessary callback methods but can designate any arbitrary method as a callback method to receive notifications for life cycle events.

A callback listener class (using lifecycle interceptor) may also be used instead of callback methods being defined in the same bean class.

Interceptors
An interceptor is a method that intercepts a business method invocation. An interceptor method may be defined in a Stateless Session Bean, Stateful Session Bean or an Message Driven Bean. An interceptor class may also be used instead of defining the interceptor method in the bean class. OC4J also supports default interceptor.

Simple JNDI lookup of EJB
Lookup of EJB has been simplified and clients do not have to create a bean instance by invoking create method on EJB and can directly invoke a method on the EJB.

Support for Web Services metadata with EJB 3.0
Web services metadata API defined under JSR-181 simplifies development of web services using metadata annotation. Oracle Application Server 10g (10.1.3.1) fully supports deployment of EJB 3.0 stateless session bean with web services metadata.

EJB 3.0 Java Persistence API
Oracle Application Server 10g 10.1.3.1 has complete support for EJB 3.0 JPA with TopLink Essentials as the default persistence provider.

TopLink Essentials derived from Oracle TopLink is the Reference Implementation for EJB 3.0 Persistence API and open sourced in Sun’s Glassfish project. TopLink Essentials is the reference implementation for EJB 3.0 JPA specification and is fully compliant.

Simplified Entities (POJO Persistence)
EJB 3.0 greatly simplifies entity beans and standardizes POJO persistence model. Entity Beans are concrete Java classes and do not require any interfaces. The entity bean classes support polymorphism and inheritance. Entities can have different types of relationships and container managed relationships are manually managed by developers.

Entity Manager API
EJB 3.0 introduces EntityManager API that is used to create, find, remove and update entities.

The EntityManager API introduces the concept of detachment/merging of entity bean instances similar to Value Object Pattern. An bean instance may be detached.
and may be updated by a client locally and then sent back to the EntityManager to be merged/synchronized with the database.

Oracle Application Server 10.1.3.1 has complete support of EntityManager API both inside and outside the EJB container.

**Metadata Annotations**
Metadata annotations greatly simplify development of entities by removing the requirement of deployment descriptors. The Entity annotation is used to specify a class to be an entity bean. Annotations are used to specify transaction attributes, security permissions, callback listeners and annotated queries.

Oracle Application Server 10.1.3.1 supports O-R mapping specified using metadata annotations and descriptors.

**Query Language Enhancements**
EJB 3.0 greatly improves the query capability for entities by using Java Persistence Query Language (JPQL). JPQL is an extension of EJB-QL that has support for additional operations such as bulk updates and deletes, JOIN operations, GROUP BY HAVING, projection and sub-queries. Also dynamic queries can be written using JPQL. Oracle Application Server 10.1.3.1 fully supports JPQL.

**Enhanced Lifecycle Methods and Call Back Listeners**
Callback listeners are supported with entities. These callback methods can be either specified using Annotations or deployment descriptor.

The following table summarizes the EJB 3.0 features supported in OC4J 10.1.3.1

<table>
<thead>
<tr>
<th>Session Beans</th>
<th>Java Persistence API</th>
</tr>
</thead>
<tbody>
<tr>
<td>• POJO bean class - No requirement for implementing javax.ejb.SessionBean interfaces</td>
<td>• Light weight persistence model</td>
</tr>
<tr>
<td>• POJI interface - No requirement for extending javax.ejb.EJBObject or javax.ejb.EJBLocalObject interfaces</td>
<td>• No component interfaces</td>
</tr>
<tr>
<td></td>
<td>• EntityManager API</td>
</tr>
<tr>
<td></td>
<td>• Metadata Annotations for O-R Mapping and O-R XML</td>
</tr>
<tr>
<td></td>
<td>• Default Mappings</td>
</tr>
</tbody>
</table>
• Enhanced lifecycle methods and call back listener classes
• Interceptor methods and classes
• Step wise migration for EJB 2.1 Session Beans to EJB 3.0
• JDK 1.4 support with XML deployment descriptors (except Interceptors)

Message Driven Beans
• POJO bean class - No requirement for implementing javax.ejb.MessageDriven interfaces
• Metadata Annotations
• Dependency Injection for EJBContext, Resources and environment references
• Enhanced lifecycle methods and call back listener classes
• JDK 1.4 support with XML deployment descriptors (except Interceptors)

Summary
Oracle Application Server 10g (10.1.3.1) provides the support for latest and leading edge Enterprise Java Beans technology to build scalable and robust service-oriented applications.