



pointing the way

White Papers

EJB 3.0

Enterprise JavaBeans



Introduction

This paper describes an Enterprise JavaBeans specification in version 3.0. It includes a comparison of previous EJB specification in version 2.1 with the newest one version 3.0.

Background

EJB is a Java technology designed for enterprise applications. Sun Microsystems defines Enterprise JavaBeans as component based architecture designed to implement distributed business applications. EJB systems are:

- ✓ Scalable,
- ✓ Support transactions,
- ✓ Offers security in multiuser environments
- ✓ Once written run on many platforms that support EJB specification.

Main part of EJB architecture is EJB Container that defines an executable environment for EJB components; it integrates enterprise services and mechanisms. Rest of EJB architecture contains mechanisms and services such as: persistence management, message-oriented programming and connector architecture.

First EJB specification was announced at JavaOne in 1998, Sun's Java developers conference. All EJB versions before 3.0 (released in 2006) share common issues that made EJB a hard to learn and use technology. For example, preparing JavaBeans to run in manageable EJB Container required from developers using variety of demanding API. That leads to significant waste of time just for handling mechanisms that simply have nothing to do with business logic. Approach introduced in earlier EJB version was too complicated to efficiently create enterprise applications.



Complications introduced by early EJB specifications:

- ✓ Creating a lot of similar code for methods that were needed by EJBs mechanisms interfaces,
- ✓ Every application required an XML descriptor that integrates the application with given environment and container's services, access to environment mechanisms was too complicated and not so intuitive
- ✓ Persistence mechanism structure made creating an application domain model unnecessary complicated.

Main goal for the newest version of EJB (that is 3.0) is simplification of an application development process including solving problem mentioned above.

Solution

Enterprise JavaBeans in the newest version 3.0 was released in 2006 and it decreases the number of programming artifacts for developers to provide, minimizes callback methods required to be implemented. Complexity of the entity bean programming model and O/R mapping model was reduced to minimum.

Annotations

Annotations were introduced in Java 5 and they represent an extra meta-information asserted to source code in the form of comment. It can be used with packages, classes, methods, parameters and attributes. Metadata represented by annotations are usually part of other source code activities. Annotations do not affect program semantics but during compile time or runtime some tools can inspect annotations to enforce desired runtime behavior. EJB version 3.0 is using this new mechanism to ease programmer's work and clarify beans declarations.

Persistence

EJB persistence mechanism is an abstraction over JDBC interface and is called Java Persistence API (JPA). Its main features are:

- ✓ Java Object and Advanced Object Oriented Concepts,
- ✓ Transactional Integrity and Concurrency,
- ✓ Large Data Processing,
- ✓ Queries Language
- ✓ Simplicity,
- ✓ Relying on a strict specification (makes it vendor independent).

JPA provides query language called JPQL that is based on traditional SQL but supports object oriented queries. Java Persistence API uses POJO and annotations to define entities and Object to Relational Mapping (ORM).

Enterprise JavaBeans Persistence mechanism starting from EJB 3.0 specification was moved to separate specification called Java Persistence API. Changes in persistence mechanism make JPA to run not only with EJB applications but also can run outside EJB Container (i.e. Java SE platform).

Message-Oriented Middleware (MOM)

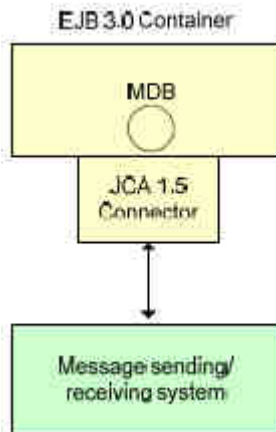
EJB 3.0 integrates with MOM functionality including RMI and asynchronous messages support. Java Message Service (JMS) and Message-Driven Beans (MDB) are used in EJB specification to send and receive asynchronous messages.

Java EE Connector Architecture (JCA)

EJB 2.0 specification introduced Message-Driven Beans – new components that are a specialized version of JMS components. MDB can easily



cooperate with others EJB component. EJB 2.1 extended MDB to support other communication standards. Today, EJB 3.0 JCA implementation is so universal that can work with many protocols such as SNMP, SMTP, BEEP. EJB 3.0 allows sending messages to EJB application – this solution provides much better integration between two systems than before.



Message-Driven components and JCA connector according to EJB 3.0.

Web Services

Web Services are network applications that use Simple Object Access Protocol (SOAP) and Web Service Definition Language (WSDL).

SOAP

This is an application protocol used by Remote Procedure Call (RPC) and asynchronous messaging systems. It is very flexible and extendable.

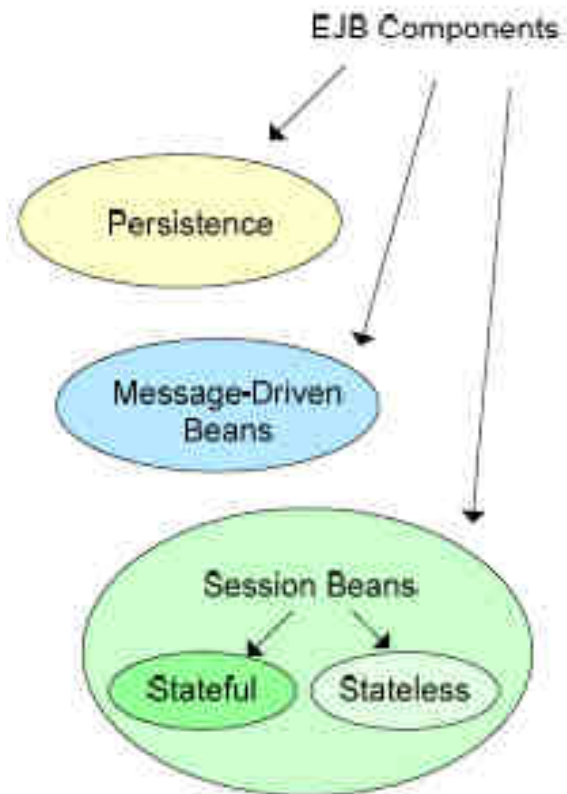
WSDL

WSDL is a language for defining web services interfaces including input and output format data, communication protocol and internet address of service.

EJB 3.0 provides access to business components using Web Services that base on JAX-WS API.

EJB Components

EJB defines three major types of beans: persistence, session and message-driven.



Three main types of EJB components.

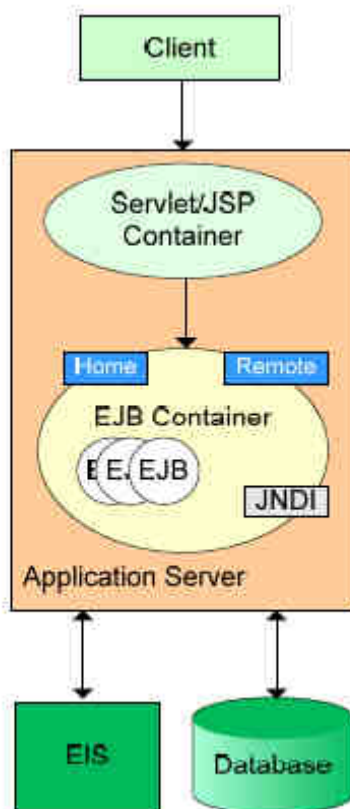
All EJB beans are a POJO that contains annotations. To define for example a session stateful bean use `@javax.ejb.Stateful` annotation before class declaration. EJB Container will recognize such POJOs as stateful bean and manage its behavior according to session stateful lifecycle.

- ✓ Persistence beans are used for managing data stores (i.e. relational databases).
- ✓ MDB integrate the EJB application with other EJB applications or messaging systems as JMS. MDB are neither persistent nor stateful.
- ✓ Session beans extend client's applications and represents processes and workflow of other EJB components.



EJB Container

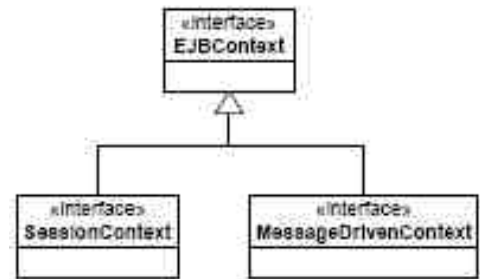
EJB Container is a main part of EJB architecture and it defines a runtime environment for Enterprise JavaBeans.



Typical JEE architecture with client layer, presentation tier (servlet container), business layer (EJB Container), Enterprise Information System and persistence layer.

EJB Container is like mediator between JavaBeans and application server.

EJB Container manages JavaBeans components, manages transactions, executes security procedures, provides concurrency mechanisms and naming service. At application startup, EJB Container provides implementation for `javax.ejb.EJBContext` interface. Every session bean and MDB has to declare `EJBContext` interface to be manageable by EJB Container.



Session bean declares descendant interface `javax.ejb.SessionContext` and MDB declares descendant `javax.ejb.MessageDrivenContext`. Approach described above lets JavaBeans to be aware of EJB environment.

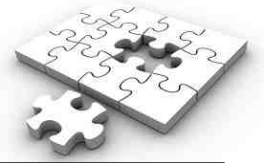
Enterprise Naming Context (ENC)

Every EJB Container that runs on applications server has internal naming context called Enterprise Naming Context. This context is implemented by Java Naming and Directory Interface (JNDI). ENC is a place where EJB Container stores JEE services' addresses.

Conclusion

New EJB 3.0 implementation: avoided using EJBHome design pattern, improved EJB Container mechanisms responsible for creating component references and searching over JNDI, session components are now plain, old java objects (POJO) that can implement just a business interfaces. EJB 3.0 is easier with fewer program artefacts and intuitive defaults.

These and some others improvement in EJB standard made from it a great technology ready to use in development of enterprise applications. Learn on earlier EJB versions mistakes, EJB 3.0 specification attracts professional java developers to study it and ease development process of JEE applications.



Links

Bill Burke, Richard Monson-Haefel – “Enterprise JavaBeans 3.0”, O’Reilly 2007

Enterprise JavaBeans Technology, EJB 3.0 Specification
<http://java.sun.com/products/ejb/>

EJB 3.0 Tutorial
<http://docs.jboss.org/ejb3/appserver/tutorial/index.html>

EJB 3.0 in a nutshell

<http://www.javaworld.com/javaworld/jw-08-2004/jw-0809-ejb.html>

Differences between EJB 3.0 and EJB 2.1

<http://help.eclipse.org/stable/index.jsp?topic=/org.eclipse.jst.ejb.doc.user/topics/cejb3vejb21.html>