163. Enterprise JavaBeans

Version 2.1.2

This course offers the Java programmer a grounding in the Enterprise JavaBeans (EJB) architecture and the skills to develop EJBs effectively for enterprise development efforts. EJB's position at the heart of the Java Enterprise platform and the use of EJB application servers as the backbone of large-scale distributed systems are studied. We consider the advantages of the application server architecture -- transaction control, security, persistence, scalability through pooling and clustering -- and study the development process for entity and session beans in depth. The J2EE reference implementation is used for all demo and lab work, and we emphasize portable EJB 2.1 code.

The focus for the first module is on end-to-end connectivity. The module follows a path roughly from the data layer to the presentation layer, so we look at entity beans first, and work demos and exercises in both Bean-Managed and Container-Managed Persistent Beans. Then the EJB session layer is considered, and both stateless and stateful session beans are developed. As part of the lab work, these are hooked to provided JSPs to illustrate the complete system and typical architecture.

*NOTE: Our JSP and Servlets courses are excellent companions to this course, and in fact the primary lab track for this module is an extension of the JSP module's labs.*

Students then proceed to the second module, which begins with some 2.x-specific features, such as message-driven beans and the features of 2.x container-managed persistence. Two EJB applications are connected via JMS messaging, one sending messages with raw JMS code, and one using a message-driven bean to receive and handle those messages by updating an accounting database. Design implications of CMP are discussed, as are the new capabilities for EJB design: home and select methods.

Students then move on to study declarative and programmatic transaction control. The EJB security architecture is also considered, and a simple role-based authorization design applied to the course's main lab project. A short chapter on exception handling in EJB clarifies the standards for exception propagation and transaction control.

The next chapter introduces the new features in EJB 2.1 for implementing
SOAP-based Web services, using the JAX-RPC mappings between Java and WSDL. Also new for 2.1, there is a short chapter on using the EJB timer service. The course concludes with a chapter on best-practice EJB development: optimizations, design patterns and implementation techniques.

The course software also includes an optional overlay of workspace and project files to support use of the Eclipse IDE in the classroom. (This requires that the instructor be experienced in use of Eclipse and able to walk students through basic tasks in the IDE.)

**Prerequisites**

- Solid Java programming experience is a must. Our Java Programming Course 103 is good preparation for this module.
- Some experience with distributed systems development, especially object-based systems such as Java RMI, CORBA, or COM is a plus.
- Some knowledge of JDBC is helpful in understanding the EJB persistence model. Understanding of XML is a plus but is not needed.
Learning Objectives

- Understand the role of Enterprise JavaBeans in enterprise-level systems development, and its relationship to other J2EE technologies such as JSP, servlets, JMS, CORBA, and XML.

- Understand the EJB architecture: the role of the EJB container in mediating contact between the client and the bean, transaction control, authorization control, and the importance of object pooling.

- Understand the development cycle for EJBs: Java source code and compilation, XML deployment descriptors, EJB compilation and deployment, and use by an application server.

- Understand the role of entity beans, session beans, and message-driven beans: their lifecycle and interactions with the container.

- Develop and test BMP and CMP entity beans and understand the importance of each of the entity-bean methods in assisting the container in pooling.

- Develop and test stateless and stateful session beans and effectively manage passivation/activation cycles.

- Use the bean context interfaces to assist with persistence code and to correctly establish bean-to-bean communication.

- Use 2.x container-managed persistence and make intelligent design choices based on the practical implications of BMP and CMP features.

- Implement 2.x CMP entity bean relationships.

- Understand the EJB transaction architecture and apply transaction attributes to EJB systems, using both declarative and programmatic approaches.

- Understand the EJB security architecture and apply roles and authorization attributes to EJB systems -- again, using both declarative and programmatic techniques.

- Implement SOAP-based Web services using stateless session beans and JAX-RPC, and understand the relationship between JAXM and message-driven beans.

- Create and respond to EJB timers in bean code.

- Tune EJB application servers and EJB code for performance.

- Understand and apply design patterns that are important to common
distributed systems and EJB systems in particular.

**Timeline: 5 days.**

**IDE Support: Eclipse 3.2**

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse 3.2. Students can code and build all exercises from within the IDE. Final build and deploy are done from the command line using Ant. Testing is done either using Ant or from the local browser via a web interface. See also our orientation to Using Capstone’s Eclipse Overlays, and please be advised that this is an optional feature; it is not a separate version of the course, and the course itself does not contain explicit Eclipse-specific lab instructions.
Module 1. Introduction to EJB

Chapter 1. The EJB Architecture

- EJB and the Java Enterprise Platform
- EJB and CORBA
- EJB Costs and Benefits
- What is an EJB?
- Entity, Session, and Message-Driven Beans
- EJB Roles
- The EJB Container
- The Home and EJB Objects
- EJB Development
- Remote Interface
- Home Interface
- Bean Class
- Remote, Local, and Web-Service Component Interfaces
- Multiple Maintenance Points
- Using JNDI in EJB

Chapter 2. Entity Beans: Bean-Managed Persistence

- Persistence Mechanism
- Instance versus Incarnation -- Similarities to the CORBA PSS
- EntityBean Interface and Responsibilities
- Entity Bean Interactions, Lifecycle and State Transitions
- Primary Keys
- EntityContext Interface and Discovering the Primary Key
- Creation and Removal
- Load and Store
- Finder Methods
- Eager vs. Lazy Persistence

Chapter 3. Entity Beans: Container-Managed Persistence

- Two Contracts for Persistence
- Declaring a Data Source
- Declaring Persistent Fields
- Implementing Finder Methods
- What the Container Does
- 1.1 Architecture
2.0 Architecture
CMP Limitations

Chapter 4. Stateless Session Beans

- Session Beans and the Presentation Layer
- Stateful vs. Stateless Sessions
- Stateful Session Bean Interactions, Lifecycle and State Transitions
- Stateless Session Bean Interactions, Lifecycle and State Transitions
- Finding Entity Beans
- Passing Value Objects

Chapter 5. Stateful Session Beans

- Stateful Session Beans
- Object Pooling and Passivation/Activation
- Activation and Passivation
- Choosing Between Stateful and Stateless
Module 2. Effective EJB

Chapter 1. Message-Driven Beans

- Asynchronous Messaging
- The Java Message Service
- Message-Driven Beans
- Lifecycle of a Message-Driven Bean
- Implementing Message-Driven Beans
- Messages and Transactions
- Message-Driven Beans and Transactions

Chapter 2. Persistence

- BMP and CMP
- Container-Managed Relationships
- EJB Query Language
- Declaring Finder Methods
- Declaring Select Methods
- Calling ejbSelect Methods from Bean Code
- Home Methods
- Design Considerations

Chapter 3. Security

- The Need for Security
- Authentication
- Authorization Model
- Declarative Authorization and Roles
- Abstract Roles and Application Assembly
- Actual Roles and Deployment
- Programmatic Authorization
- Identifying the Caller

Chapter 4. Transactions

- The Need for Transactions
- ACID Properties
- Declarative Transactions
- Transaction Attributes and Isolation Levels
- Scenarios
Programmatic Transactions
EJB Context and Monitoring the Transaction
SessionSynchronization Interface

Chapter 5. Web Services and EJB

SOAP-Based Web Services
WSDL
JAX-RPC
EJB 2.1 and JAX-RPC
Session Beans as Web Service Endpoints
How It Works -- Build Time and Runtime
The Bean's Service Endpoint Interface
SOAP as an RMI Transport
Adding a SOAP Interface to a Session Bean
Generating From WSDL
"Gotchas"
J2EE and Web Services

Chapter 6. Exception Handling

Java Exceptions
Remote Exceptions
EJB Exceptions
Application Exceptions
Propagation to the Client
System Exceptions
Automatic Transaction Rollback

Chapter 7. The Timer Service

Uses for Timers
The EJB Timer Service
Service, TimedObject and Timer Interfaces
Timer Handles
Creating a Timer
Implementing ejbTimeout
Timers and Transactions

Chapter 8. Best Practices

The Bean Environment
Expressing UML Designs for EJB
Creation and Finder Semantics
Designing for Latency
Value Object Pattern
Local vs. Remote Beans and the Façade Pattern
Transaction Models
When Not to Use EJB
Flyweight Pattern versus EJB Pooling
Entity Bean Performance Optimizations

Appendix A. Learning Resources

System Requirements

Hardware Requirements (Minimum)  500 MHz, 256 meg RAM, 500 meg disk space.
Hardware Requirements (Recommended)  1.5 GHz, 512 meg RAM, 1 gig disk space.
Operating System  Tested on Windows XP Professional. Course software should be viable on all systems which support a J2SE 1.4 SDK.
Software Requirements  All free downloadable tools.