

JBoss for Advanced J2EE Developers

This is our advanced training targeted at advanced developers, developers wanting to participate in the JBoss Professional Certification program and team leads/architects who want to base their applications on JBoss. Divided into 80% theory and 20% labs, this course is intensive, high-level, theoretical and in-depth. A good deal of lecture time is spent understanding the fundamental design choices behind JBoss, the Aspect Oriented Programming nature, the detyped invocation flow the abstract interceptor on the client and server side, and how easily you can apply all this to build your own strand of JBoss optimized for your needs.

Course Duration: This is a 4 day course.

Course Format: Divided into 80% theory and 20% labs, this course is intensive, theoretical and in-depth.

Note: This course is not recommended for beginner Java programmers.

Course Prerequisites

Completion of the JBoss for Java Developers or equivalent knowledge through experience with developing J2EE applications is required. It is recommended that the student have a minimum of 1 year practical experience in developing J2EE applications specifically, it is suggested that the student has been an active developer on a comprehensive multi-tier J2EE application that has been deployed on the JBoss Application Server.

The developer should have practical experience with each of the following topics:

- JNDI
- JDBC
- Servlets and JSPs
- Enterprise Java Beans
- JMS
- The J2EE Security Model
- J2EE application development and deployment on the JBoss Application

While prior knowledge of JMX is helpful, it is not required. This training is based on the 4.x series of the JBoss Application Server.

Course Modules

- 1. JBoss State of the Union.** In this brief introduction we present JBoss the Open Source project and JBoss Inc. the provider of Professional Open Source services. We establish the current status of the project and cover the product roadmaps planned for the next JBoss product releases. We will tell you where we are today, and where we plan to go tomorrow.
- 2. JBoss Architecture Overview.** The JBoss architectural overview paints the full picture of the app-server microkernel design -- the JMX invocation bus, MBeans as services and containers. We briefly focus on the containers and then back some more to the unified classloader architecture, before getting to the multiple detached invoker layer and the client proxy tricks. This module is a must requirement for any systems architect looking to understand the modular nature of the application server, how to take advantage of customized solutions with JBoss and where to find the extensions point to mold the server to fit your application's needs.
- 3. JMX in JBoss.** In this module we go in depth into the JMX specification, standard/dynamic and model MBeans covering the XMBeans implementation of JBoss. This focuses on the usage of JMX and where we apply it in the application server. You will learn how to create management interfaces for your service implementations, how to decouple your services running in the application server and how to externalize the service definitions. We will show how to take advantage of additional JMX service such as persistence in this section. Core and fundamental knowledge to any developer looking to build customization and extensions to the application server.
- 4. JBoss Microkernel.** Creating extensions and new modules is easy with the JMX based service definition in the JBoss application server. This module explains how you integrate your own custom modules into the JBoss microkernel. We cover the deployer architecture and implementation, classloader architecture and how to integrate your custom classloaders into the JBoss classloading domains. We will show you how we bootstrap the application server with practical tips on how to use the JMX API and most importantly, how to implement your own service implementations and have them participate as first class application server modules in the kernel lifecycle. Anyone wishing to extend the application

server to fit their specific application requirements must understand how to take advantage of the kernel integration.

5. **JBoss Smart Proxies.** In this section you will fully understand how the dynamic proxy approach of JBoss application server works. Learn how to customize the proxy implementations executed by the client applications, how to take advantage of the location dependent information inside the proxy and how to customize the on-the-wire payload sent by the client to the application server. For developers looking to extend the transparent wire protocol of the proxy with additional security or identity context, understanding the smart proxy architecture is vital.
6. **EJB Container.** In this module we will walk you through the EJB container implementation. We will show how the interceptor based middleware design allows your server to be configured as an EJB container, how each of the individual interceptors implements a specific aspect of the EJB specification, and how an EJB invocation travels through the interceptor stack to execute the bean logic. Essential information to anyone who needs to understand how to interact with the abstracted invocation object, how to implement interceptors on the server side, how to integrate with different server side services, such as transaction manager and security manager, and how to customize or optimize the EJB container itself.
7. **Introduction to EJB3.** This module introduces you to the next generation of J2EE application development using the EJB3 component model. We will cover the major new features of the EJB3 specification and show you how the new component model allows middleware developers to build their applications with regular Java objects. Java 5 annotations, POJO based EJBs, EJB callbacks and EJB specification defined interceptors will be introduced in this module. We will also introduce the major new features to EJB persistence, new POJO based entity beans and persistence with inheritance and polymorphism. This module gives you a head-start in understanding what the new specification brings to the table and how it affects future EJB deployments.
8. **Aspect Oriented Middleware.** This brand new module takes you to the future of Java middleware development. Expanding on the topics covered with earlier sections, we generalize the interception architecture and define JBoss AOP framework, and show you how to build your custom containers from scratch in a robust and predictable manner. We explain basic AOP concepts such as aspects, pointcuts and bindings and show how to use those in practice to implement a POJO based middleware applications. We will also cover how to apply existing EJB related services (security, transactions, locking, etc) on plain

old Java objects, and how to turn your existing domain objects into fully blown containers without imposing disruptive component contracts to the implementation.

- 9. JBossCache.** Learn how to use and take advantage of industrial strength distributed cache implementation from JBoss. JBoss Cache is a replicated and transactional cache that allows you to cache data from Enterprise Applications in order to dramatically improve performance. The caching data is replicated automatically allowing you to easily cluster data across a grid of JBoss servers. JBoss Cache can run as an MBean service within JBoss Application Server or other J2EE containers. We cover both the TreeCache implementation and the AOP enhanced cache implementation – features and automatic object mapping (with AOP TreeCache) on the cache tree structure. How eviction policies work and how the cache interacts with persistent storage. Important knowledge for anyone looking to provide highly scalable and responsive systems with high availability and full transactional support.
- 10. EJB Clustering.** In this module we cover the current clustering implementation for EJB implementations, from replicated JNDI tree implementation to load balancing and failover features of EJBs. We will show you how we take advantage of the JBoss smart proxies with pluggable load balancing implementations and how the proxies are dynamically updated with cluster topology changes.
- 11. Fine Tuning JBoss Application Server.** Learn the tricks and skills to fine tune your JBoss application server for production usage. Find out what to do to minimize serialization, how to increase your server's concurrency and throughput with correct locking policies and how to make an effective use of caching for your EJB's. We cover the differences between default pessimistic, instance per transaction and optimistic concurrency. Cache invalidation strategies and commit options, fine tuning the virtual machine and read-mostly clustering strategies. Critical knowledge before going to production with your JBoss application server!
- 12. Security in JBoss.** Learn how to secure your EJB applications with JBoss. This module explains the nuts and bolts of JBoss' security implementation, how you can configure it to integrate with your existing security infrastructure and how to extend it to do your own custom security implementations. We will discuss how to use JAAS to integrate with existing databases and LDAP authentication systems, how to do per-instance security based on EJB state and invocation context, and how the security context propagates between components in the application server.

13. JBossMQ. This module covers our messaging implementation. We will see an architecture overview into JBossMQ implementation, the different parts of the messaging infrastructure and how they interact with each other. We will show you how the message passes through the system, what the different invocation layers are, how to create your own server side interceptors for messaging, how to secure your JMS destinations and how to configure your message persistence and cache. Learn all about the high availability features of JBoss messaging implementation and how to use and configure it. For anyone using messaging in their application, this module provides all the essentials.