

JBoss jBPM

JBoss jBPM training is targeted toward system architects and developers who work closely with business analysts and are responsible for bringing business processes into J2EE environment using jBPM as a BPM engine. In addition, The JBoss jBPM training will provide students with a thorough understanding of the BPM landscape, types of engines and positioning of the buzzwords.

Students will acquire practical hands-on expertise and will be ready to start developing business processes with JBoss jBPM after the course. Another goal of the training is to provide a thorough preparation for comparing workflow engines.

Course Duration: This is a 3 day course.

Course Format: The structure of the course is 60% theory and 40% hands-on lab exercises.

Course Prerequisites

The student must have the following skills:

- The student must have previous experience developing an Hibernate application. The student must know how to configure a simple SessionFactory for Hibernate, utilize a Hibernate Session and transactional demarcation and how to perform basic queries on Hibernate objects.
- Competency with Java application development.
- Previous exposure to the concepts of workflow and business process modeling (BPM) is **not** required
- Experience with JBoss Eclipse or the Eclipse IDE with the JBoss plugin is recommended but not required
- Basic notions of JUnit test framework is recommended.

Course Modules

- 1. jBPM Overview.** We start by explaining the BPM landscape of workflow and orchestration. The goals of jBPM and how it fits into the overall software development process are discussed. All the buzzwords are covered in depth as well as their relation. We address Graph Based Execution Languages and Business Process Management. The student is provided with a brief introduction to Graph Oriented Programming and an overview of the jBPM components.
- 2. Basic Process Modeling.** A process language is usually just a set of node-types or workflow constructs that can be used in a graphical designer. JBoss jBPM has an open ended model and already supports two process languages: JPDL and BPEL. This module covers the JPDL nodes that are available for modeling your processes with jBPM. It also covers process/subprocess information transfer, process/subprocess bindings, actions and concludes with the student installing the jBPM software.
- 3. The Graphical Process Designer.** In this module the student is shown how to define a business process using the Graphical Process Designer (GPD). The module concludes with a lab allowing the student to create process for web-based sales.
- 4. Deployment.** In this module the student will learn the various ways in which a jBPM process can be deployed and how different versions of the same process can be concurrently deployed and used. The student will also learn how to deploy the jBPM run-time, deploy a Hibernate-supported database for a jBPM application and deploy the jBPM identity component.
- 5. Client Programming.** In this module the student will learn, through six highly-applicable use cases, the jBPM client API and the associated jPDL. The use cases include a simple process definition and execution, storing a process instance in the jBPM database, working with process variables without persistence, a simple task assignment, creating custom actions and using jBPM with a Stateless Session EJB wrapper.
- 6. Advanced Process Modeling Concepts.** This module will begin with a brief high-level review of topics covered previously such as the process definition, node execution and node types. From this point of commonality the student will learn how to augment the process with behaviors associated with events, actions and transitions throughout the process graph. The student will learn how to incorporate superstates, use exception handling, process composition, create

custom node behavior and control graph execution. The student will be introduced to Sync/Async jBPM and how to establish transaction demarcation.

- 7. Integration.** This module is all about integrating a jBPM process into an application design or, put another way, how to communicate with jBPM. The student will learn how to program using external triggers and action handlers.
- 8. Pluggable Architecture.** The jBPM library is extremely capable “out-of-the-box”, however, jBPM’s capabilities can also be programmatically extended. In this module the student will learn how to add custom modules, extend the task instance and adding new events to the event system.
- 9. Persistence.** Without persistence, the execution state of a process could not be stored and later retrieved. This module covers the jBPM API to store, retrieve and update process executions in the database. It covers caching and how to integrate jBPM database into transactions of your environment. The jBPM database updates can be combined in an applications transactions, or alternatively, in global transactions such as the sending of asynchronous messages. The student will learn about the different forms of process data, the underlying data base factories, the jBPM context and its component Hibernate Sessions. The student will be guided through 3 use cases, each presented to show the programmatic capabilities available via the jBPM context API. The module concludes with addressing advanced Hibernate topics and the use of clustering with jBPM.
- 10. Context.** Process variables contain the information that is maintained for the lifetime of the while process execution. Since this lifetime may span wait states, also the process variables need to be persistable. In this module we’ll cover the API’s manipulate the process variables and show how to customize the way that the process variables are stored in the database.
- 11. Task Management.** Management of tasks for people is a crucial capability of the JBoss jBPM product. First of all we take a look at the relation between the process and the tasks: the task-node. Second, the assignment of tasks to people will be explained. Third, the optional jBPM organization component will be covered that allows for support of group assignments and runtime actor expressions. As a final subtopic, we’ll provide the concrete pointers to integrate the JBoss jBPM task lists with other applications.
- 12. Web Forms.** The web forms mechanism is an extension of the JBoss jBPM web application console that allows process developers to create a web interface for process participants in no time. This is extremely useful for rapid prototyping and pilot projects. The process developer needs to add minimal information about

the data in each task in the process. Step by step we'll show what information needs to be added and how this translates into the web application.

13. Introduction to BPEL. As the name states, this module provides a brief introduction to BPEL. The student will be provided the concept of, motivation behind and the relationship of the BPEL standard to other standards. The functional aspects of BPEL are covered as well as a comparison of BPEL to jPDL and why one language may be more appropriate than the other in specific situations. The module concludes with showing the student how to deploy a process defined in BPEL to the jBPM engine.