Configuration Management with Maven

Duration:	2 days
Type:	intermediate

Description

This course enables experienced Java developers to build and deploy JSE and JEE applications using the Maven configuration management tool. Delegates install Maven from scratch and use it to build client side, single-module server side and multi-module server side applications.

Initially delegates use Maven from the command line, but quickly progress to using it from within the Eclipse IDE. In order to accurately reflect real world usage the sample projects have dependencies on multiple frameworks, including testing tools, web frameworks and persistence frameworks. The course can be customized to use those frameworks in use within the clients organisation.

Prerequisites

Delegates must be accomplished Java developers who have experience of building and deploying JSE and JEE applications. Additionally they must be familiar with all of the frameworks selected for use in sample applications (see above). Experience of Ant or Make is very useful but not essential.

List of Modules

A Review of Java Deployment (Optional)

- Creating JAR files with jar.exe
- Viewing and editing manifest files
- Deploying an application in a JAR file
- The structure of a JEE Web Module
- Deploying Web Applications in WAR files
- The structure of a JEE EJB Module
- Deploying Enterprise Applications

Introducing Maven

- The evolution of the Maven build tool
- The architecture and philosophy of Maven
- Comparing Maven to Ant and Make
- Downloading and installing Maven
- Creating and running a simple project
- The default structure of a Maven project
- Using the *help* plugin to examine your project
- Integrating Maven and the Eclipse IDE

Core Concepts of Maven

- Understanding the Project Object Model (POM)
- Examining the contents of the Super POM
- Identifying a project via Maven coordinates
- Relationships between POM's and the Super POM
- How Maven manages dependencies between POM's
- The phases of the default Maven lifecycle
- How plugin goals are attached to lifecycle phases
- Examples of package types with additional phases

Building a Client Side Application

- Creating the project and customizing *pom.xml*
- Identifying dependencies and adding them to the POM
- Adding the implementation, tests and resource files
- Packaging, testing and running the application

Using Maven in Real World Projects

- Creating projects containing multiple modules
- Using multi-module projects to build JEE applications
- Viewing and optimizing the dependencies of your project
- Using the five standard scopes to limit dependencies
- Specifying dependencies using a range of version numbers
- Adding your own behaviour to arbitrary steps in the lifecycle
- Using profiles to ensure portability across different environments

Building a Server Side Application

- Declaring a multi-module project that builds an enterprise application
- Creating a web interface project using frameworks (Struts, Spring MVC, Stripes etc...)
- Creating a project for persistent classes (using iBatis, Hibernate or the JPA)
- Creating a project for controller classes (using Spring for dependency injection)
- Lifting settings and dependencies from the child to the parent POM
- Optimizing dependencies by eliminating duplication and restricting scope
- Packaging, unit testing, deploying and acceptance testing the application

Advanced Uses of Maven

- Using assemblies to create your own packaging formats
- Using resource filtering to replace values in property files
- Generating a project specific website via Maven
- An overview of writing your own Maven plug-ins