

**Introduction to Apache Maven**

**Course Number:** MVN-100WA
**Duration:** 2 days

**Overview**

Accelebrate's Introduction to Apache Maven training class teaches attendees how to automate the build of Java projects using Apache Maven. Note: The most popular version of the course is taught with Eclipse but it can be delivered with any Integrated Development Environment (IDE).

**Prerequisites**

All attendees must have core Java and Java web programming experience.

**Materials**

All attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

* JDK 8 or later
* Any operating system that supports Java 1.8 or later
* The Java tool the students are likely to use after the class (Eclipse is recommended, but other tools are also supported)
* The Maven version of your choice
* Other free software - please contact us if you have purchased this class

**Objectives**

* Download and install Maven
* Build a project
* Work with Maven's directory structure, plugins, repositories, and more
* Understand the Project Object Model (POM)
* Build a complete web application using Maven
* Build and activate profiles
* Work with popular Maven plugins
* Use Maven from Eclipse via the m2eclipse plugin

**Outline**

* Introduction to Apache Maven
	+ Build Tools for Java
	+ History of Build Tools
	+ Traditional Scripting
	+ 'make'
	+ Problems with Make
	+ Manual Build with JavaC
	+ ANT
	+ Pros and Cons of Ant
	+ Apache Maven
	+ Goals of Maven
	+ What is Apache Maven?
	+ Why Use Apache Maven?
	+ The Maven EcoSystem
	+ Consistent Easy-to-Understand Project Layout
	+ Convention Over Configuration
	+ Maven is Different
	+ Maven Projects have a Standardized Build
	+ Effect of Convention Over Configuration
	+ Importance of Plugins
	+ A Key Point on Maven!
	+ Key Features of Maven
* Installing and Running Apache Maven
	+ Downloading Maven
	+ Installing Maven
	+ Run From Command Line
	+ Running Inside an IDE
	+ Settings.xml
	+ Local Repository
* Getting Started with Maven
	+ Terminology and Basic Concepts
	+ Artifacts
	+ Lifecycle
	+ Default Lifecycle
	+ Plugins
	+ Running Maven - the Story So Far
	+ Running Maven from an IDE
	+ Common Goals
	+ pom.xml
	+ Artifact Coordinates
	+ Standard Layout for Sources
* A Web Application in Maven
	+ A More Complex Project
	+ Putting it Together With Maven
	+ Packaging the Target Artifact
	+ The Source Tree
	+ Dependencies
	+ Transitive Dependencies
	+ Dependency Scope
	+ Working With Servers
	+ Declaring and Configuring Plugins
	+ Running the Plugin
	+ Binding a Plugin Goal to the Lifecycle
	+ Archetypes
* Commonly Used Plugins
	+ Maven Plugins
	+ Declaring and Configuring Plugins
	+ Running the Plugin
	+ Binding a Plugin Goal to the Lifecycle
	+ Maven Surefire Test Plugin
	+ Failsafe Plugin
	+ Site Plugin
	+ JavaDoc Plugin
	+ PMD Plugin
	+ Code Coverage – Cobertura
* Multi-Module Builds
	+ Introduction
	+ The Reactor
	+ Reactor Sorting
	+ Multi-Module Build by Example
* POM Projects
	+ Project Object Model (POM)
	+ The overall POM structure
	+ Storing POM
* Writing Maven Plugins
	+ What is Maven Plugin
	+ Example of Using a Plugin
	+ Create a Custom Plugin
	+ Plugin Management
* Creating Archetypes
	+ Introduction to Maven Archetypes
	+ Using Interactive Mode to generate Goal
	+ Common Maven Archetypes
* Repository Management
	+ Maven's Approach to Artifacts
	+ Publishing Artifacts
	+ Summary of Maven's Artifact Handling
	+ Repository
	+ Repository Manager
	+ Proxy Remote Repositories
	+ Types of Artifacts
	+ Release Artifacts
	+ Snapshot Artifacts
	+ Reasons to Use a Repository Manager
	+ Repository Coordinates
	+ Addressing Resources in a Repository
* Release Management
	+ What is release Management?
	+ Release Management with Nexus
	+ Release Management with Maven
* Conclusion