

# Continuous Integration with Maven, Jenkins, and Artifactory

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy  
Get Your Free Copy Now

**Course Number: JEN-100**

**Duration: 3 days**

## Overview

## Course Description

This Continuous Integration with Maven, Jenkins, and Artifactory training course teaches attendees the Apache Maven build process, the principles of continuous integration, and how to implement continuous integration with automated test execution using Jenkins, Maven, and the Artifactory repository manager. Students use a local copy of Jenkins to create and run Maven jobs, link to a version control system, and run automated testing and generate development reports. Attendees configure the system to log build reports and generated artifacts to an Artifactory repository.

## Skills Gained

- 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
- Use Maven from Eclipse via the m2eclipse plugin
- Install and configure Jenkins in a servlet container
- Create Jenkins builds
- Configure and use Apache Ant and Apache Maven with Jenkins
- Use Jenkins to generate Java coding standards reports, code coverage reports, and change notices
- Use Jenkins to automatically deploy software into a testing environment
- Configure Maven and Jenkins to deploy the generated artifacts to Artifactory

## Prerequisites

Attendees should have familiarity with Java development practices.

## Training Materials

All students receive comprehensive courseware.

## Software Requirements

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Ascendient Learning will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

## Audience

## Course Details

## Outline

- Introduction To Continuous Integration And Jenkins-CI
  - Agile Development
  - Agile Development (cont'd)
  - What is Continuous Integration
  - What is Continuous Integration (cont'd)
  - What is Continuous Integration (cont'd)
  - Typical Setup for Continuous Integration
  - Jenkins Continuous Integration
  - Jenkins Features

- Running Jenkins
- Introduction To Apache Maven
  - Build Tools for Java
  - Build Tools for Java (cont'd)
  - 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?
  - What is Apache Maven (cont'd)
  - 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
  - xml
  - Local Repository
- Installing and Running Jenkins

- Downloading and Installing Jenkins
- Running Jenkins as a Stand-Alone Application
- Running Jenkins on an Application Server
- Installing Jenkins as a Windows Service
- A Jenkins Job
  - Different types of Jenkins job
  - Configuring Source Code Management(SCM)
  - Working with Subversion
  - Build Triggers
  - Schedule Build Jobs
  - Polling the SCM
  - Maven Build Steps
- 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
  - xml
  - Example
  - Example (cont'd)
  - 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 Plugins (Maven)
  - What is Maven Plugin
  - Example of Using a Plugin
  - Create a Custom Plugin
  - Create a Custom Plugin (cont.)

- Plugin Management
- Creating Archetypes
  - Introduction to Maven Archetypes
  - Introduction to Maven Archetypes (cont.)
  - Using Interactive Mode to generate Goal
  - Common Maven Archetypes
  - Summary
- Repository Management
  - Introduction
  - Proxy Remote Repositories
  - 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
- Jenkins Plugin
  - Introduction
  - Jenkins Plugins - SCM
  - Jenkins Plugins – Build and Test
  - Jenkins Plugins – Analyzers
  - Jenkins for Teams
  - Installing Jenkins Plugins
- Securing Jenkins
  - Jenkins Security - Overview
  - Jenkins Security
  - Authentication
  - Authorization
  - Confidentiality

- Activating Security
- Configure Authentication
- Using Jenkins's Internal User Database
- Creating Users
- Authorization
- Matrix-Based Security
- Note – Create the Administrative User
- Project-based Matrix Authorization
- Project-Based Authentication
- Distributed Builds With Jenkins
  - Distributed Builds - Overview
  - Distributed Builds – How?
  - Slave Machines
  - Configure Jenkins Master
  - Configure Projects
- Continuous Deployment and the Jenkins Pipeline
  - Continuous Deployment
  - DevOps and Continuous Deployment
  - Continuous Deployment Challenges
  - Continuous Deployment with Jenkins
  - The Pipeline Plugin
  - Defining a Pipeline
  - A Pipeline Example
  - Parallel Execution
  - Creating a Pipeline
  - Invoking the Pipeline
  - Interacting with the Pipeline
- Best Practices for Jenkins
  - Secure Jenkins
  - Backups
  - Reproducible Builds
  - Testing and Reports

- Large Systems
- Distributed Jenkins
- Summary
- Conclusion