

Maven Fundamentals



Bryan Hansen

Director of Software Development

@bh5k



Version Check



This version was created by using:

- Maven 3.9
- Java 17

Course Overview

Introduction

Key concepts

Day-to-day coding

Integration

Complexities

Topics

Introduction to Maven

Structure

Dependencies

Repositories

Plugins

IDE Integration

Bill of Materials (BOM)

What is Maven?

Build Tool

- One Artifact
- Manage Dependencies

Project Management Tool

- Handles Versioning/Releases
- Describes Project
- Produces Javadocs/Site Information

Who Owns It?

Apache Software Foundation

Maven is built with Maven

Open Source

Why Use It?

Repeatable builds

Transitive dependencies

Environment

Local repo

IDE and Standalone

Preferred method



Ant

Replacement for Make

Cross platform

Java and XML based

Ant

```
<target name="clean" description="clean up">  
  <delete dir="${build}" />  
</target>
```

Ant

```
<project>
  <target name="delete_everything" description="clean up">
    <delete dir="build" />
  </target>
  <target name="compile_code" depends="init" description="compile the
source">
    <mkdir dir="build/classes" />
    <javac srcdir="src" destdir="build" />
  </target>
  <target name="wrap">
    <mkdir dir="build/jar" />
    <jar destfile="build/jar/HelloWorld.jar" basedir="build/classes" />
  </target>
</project>
```

Maven

Full Featured

Implicit

Consistency

Inheritance

Transitive dependencies

Versioned

Pros and Cons

Maven

Black box

Steeper learning curve

Convention over Configuration

IDE

Less Overhead

Mindset

VS

Ant

Easily trace

Quick to learn

Copy-and-paste

Large project size

Ant

```
<project>
  <target name="delete_everything" description="clean up">
    <delete dir="build" />
  </target>
  <target name="compile_code" depends="init" description="compile the
source">
    <mkdir dir="build/classes" />
    <javac srcdir="src" destdir="build" />
  </target>
  <target name="wrap">
    <mkdir dir="build/jar" />
    <jar destfile="build/jar/HelloWorld.jar" basedir="build/classes" />
  </target>
</project>
```

Maven

```
<project>  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>com.pluralsight</groupId>  
  <artifactId>FitnessTracker</artifactId>  
  <version>0.0.1-SNAPSHOT</version>  
  <name>HelloWorld</name>  
</project>
```

Gradle?

```
apply plugin: 'java'
apply plugin: 'eclipse'
apply plugin: 'application'

mainClassName = 'hello.HelloWorld'

// tag::repositories[]
repositories {
    mavenCentral()
}
// end::repositories[]

// tag::jar[]
jar {
    archiveBaseName = 'gs-gradle'
    archiveVersion = '0.1.0'
}
// end::jar[]

// tag::dependencies[]
sourceCompatibility = 17
targetCompatibility = 17
```


Summary

Ant is declarative

Maven convention

Ant is scripted

Maven project lifecycle