Maven Fundamentals

Bryan Hansen
twitter: bh5k
http://www.linkedin.com/in/hansenbryan

Course Overview

- **Introduction to Maven**
  - New to Maven
  - Ant Veteran
- **Key concepts of Maven**
  - Convention over configuration
- **Day-to-day coding**
- **Integration**
- **Maven can be very complex**

Course Overview

- **Topics Covered**
  - Introduction to Maven
  - Structure
  - Dependencies
  - Repositories
  - Plugins
  - IDE Integration

Introduction to Maven

Bryan Hansen
bryankhansen@gmail.com

Outline

- High level overview
- Ant VS Maven VS IDE
- Installation Best Practices
- Hello World Application
- Summary

Maven High Level Overview

Bryan Hansen
bryankhansen@gmail.com

What is Maven

- **At its simplest, Maven is a build tool**
  - It always produces one artifact (component)
  - It helps us manage dependencies
- **It can also be used as a project management tool**
  - It handles versioning and releases
  - Describes what your project is doing or what it produces
  - Can easily produce Javadocs as well as other site information

Who owns it

- Maven is managed by the Apache Software Foundation
- Maven sites are built with Maven
- Open Source

Why do you want to use it

- **Repeatable builds**
  - We can recreate our build for any environment
- **Transitive dependencies**
  - Downloading a dependency with also pull other items it needs
- **Contains everything it needs for your environment**
- **Works with a local repo**
- **Works with your IDE, but also standalone**
- **The preferred choice for working with build tools like Jenkins or Cruise Control**

Ant VS Maven

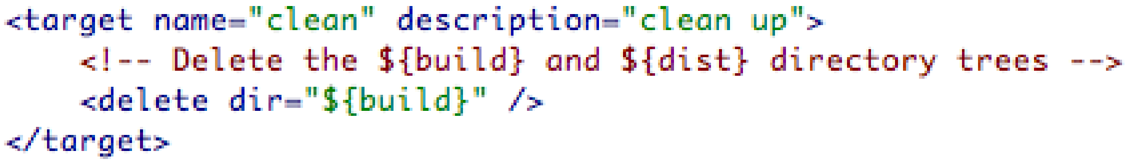
Bryan Hansen
bryankhansen@gmail.com

Ant

- Ant was developed to be a replacement for a build tool called Make
- Designed to be cross platform
- Built on top of Java and XML
- Very procedural

Ant

- **Ant really isn't a build tool as much as it is a scripting tool**
- **You have to explicitly do everything in Ant**



```
<target name="clean" description="clean up">
    <!-- Delete the ${build} and ${dist} directory trees -->
    <delete dir="${build}" />
</target>
```

- **We have to define everything that we want to do**
  - clean, clear, cleanup, etc…
- **A lot of tribal knowledge, nothing carries over**

Maven

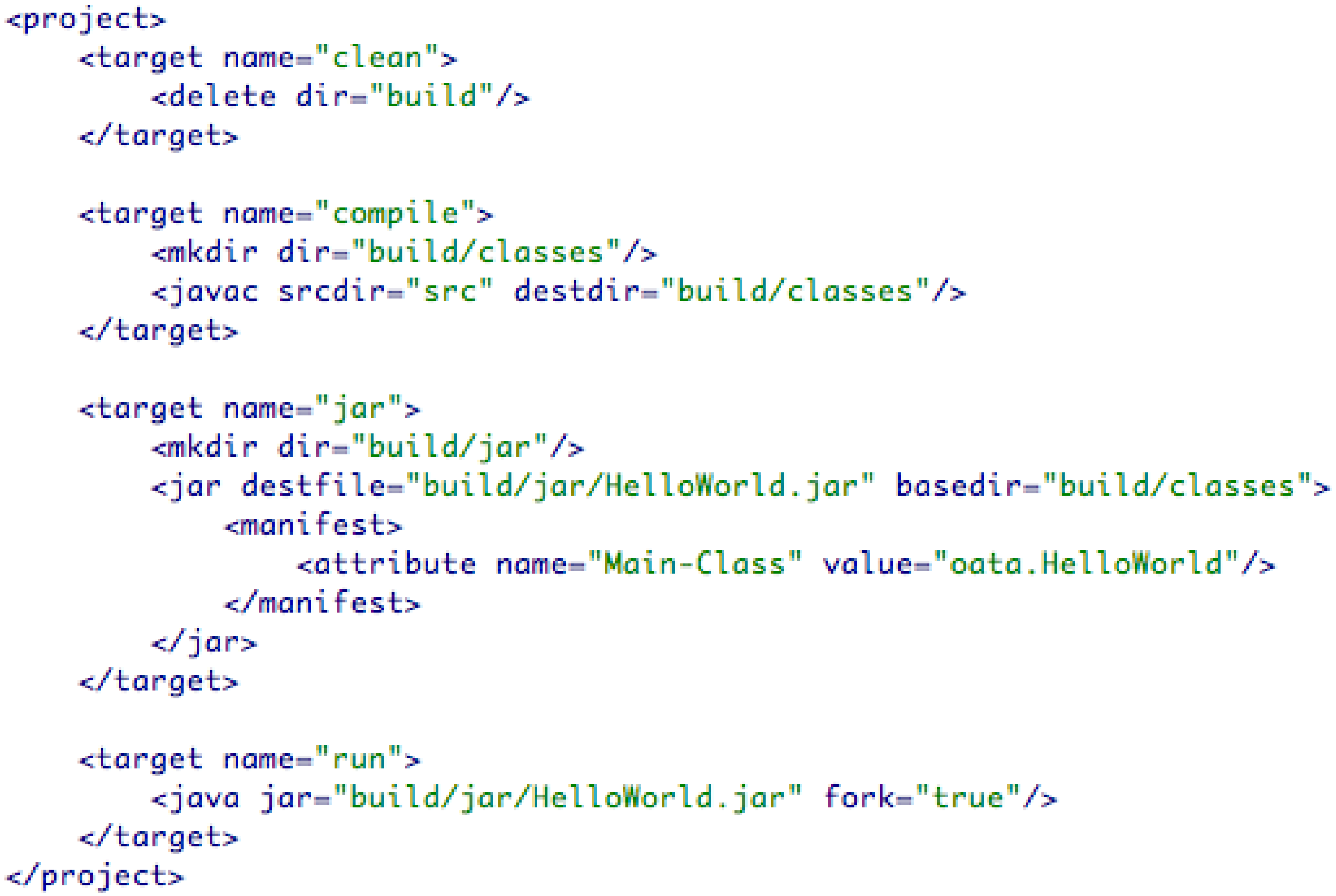
- Maven is a full fledged build tool
- A lot of implicit functionality
- Consistency across projects
- Also capable to achieve inheritance in projects
- Transitive dependencies (can be achieved using Ivy with Ant though)
- Built around versioning

Pros and Cons

- Maven can be a black box
- Steeper learning curve
- Convention over configuration
- Better IDE integration
- Let overheard through use of repos
- Different mindset, steepest learning curve is not making Maven act like Ant

- You can trace through Ant files fairly easily
- Quicker to learn, but very copy-paste intensive
- Larger project size in SCM, artifacts stored with project

Ant build.xml



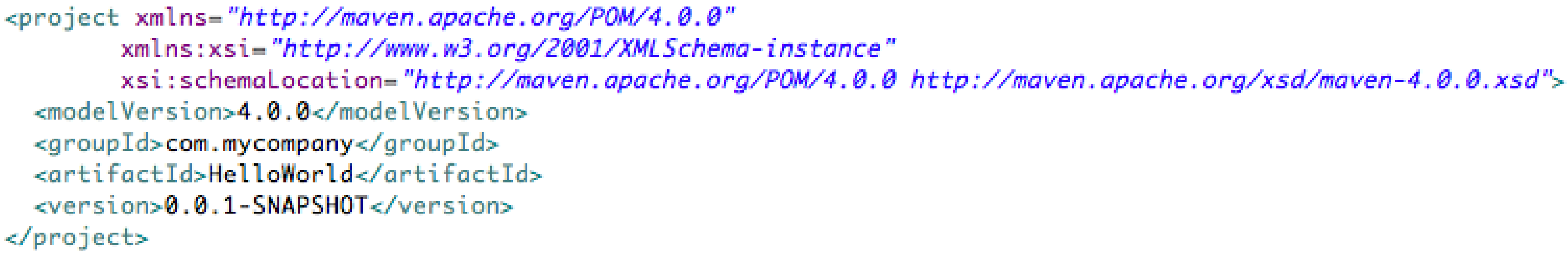
```
<project>
    <target name="clean">
        <delete dir="build"/>
    </target>

    <target name="compile">
        <mkdir dir="build/classes"/>
        <javac srcdir="src" destdir="build/classes"/>
    </target>

    <target name="jar">
        <mkdir dir="build/jar"/>
        <jar destfile="build/jar/HelloWorld.jar" basedir="build/classes">
            <manifest>
                <attribute name="Main-Class" value="oata.HelloWorld"/>
            </manifest>
        </jar>
    </target>

    <target name="run">
        <java jar="build/jar/HelloWorld.jar" fork="true"/>
    </target>
</project>
```

Maven pom.xml



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany</groupId>
  <artifactId>HelloWorld</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

Summary

- Ant is very declarative
- Maven follows a convention over configuration model
- Ant is maybe easier to learn, but it really is only beneficial as a scripting tool
- Maven is really centered around managing your entire project's lifecycle

Summary

Bryan Hansen
bryankhansen@gmail.com

pluralsight
hardcore developer training

What we covered

- **What is Maven**
- **How is Maven different from other tools**
  - Its not just a scripting tool
- **Where to get it and how to install it**
- **A Hello World application**

What is next

- Structure
- Dependencies
- Repositories
- Plugins
- Eclipse Integration