

# Plugins

Bryan Hansen

twitter: bh5k

<http://www.linkedin.com/in/hansenbryan>



# Outline

- **Goals**
- **Phases**
- **Compiler plugin**
- **Jar plugin**
- **Sources plugin**
- **Javadoc plugin**

# Goals

- The default goals are plugins configured in the maven install
  - clean, compile, test, package, install, deploy
- Super pom has these goals defined in it, which are added to your effective pom:

```
<plugin>
  <artifactId>maven-clean-plugin</artifactId>
  <version>2.4.1</version>
  <executions>
    <execution>
      <id>default-clean</id>
      <phase>clean</phase>
      <goals>
        <goal>clean</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

- Goals are tied to a phase

# Phases

- **validate**
  - Validate the project is correct and all necessary information is available
- **compile**
  - Compile the source code of the project
- **test**
  - Test the compiled source code
- **package**
  - Packages the code in its defined package, such as a JAR

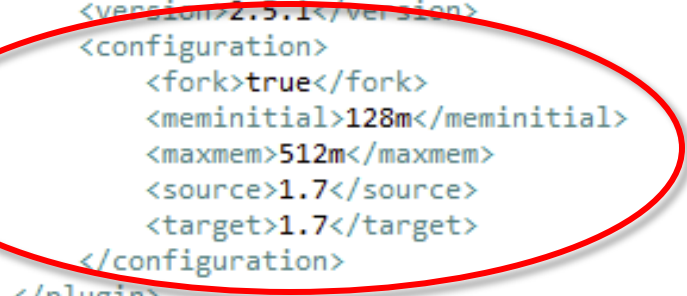
# Phases

- **integration-test**
  - Deploy and run integration tests
- **verify**
  - Run checks against package to verify integrity
- **install**
  - Install the package in our local repo
- **deploy**
  - Copy final package to a remote repository

# Compiler Plugin

- Used to compile code and test code
- <http://maven.apache.org/plugins/maven-compiler-plugin/index.html>
- Invokes Javac, but with the classpath set from the dependencies
- Defaults to Java 1.5 regardless of what JDK is installed
- Configuration section allows customization
  - Includes/Excludes
  - Fork
  - Memory
  - Source/target

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.5.1</version>
  <configuration>
    <fork>true</fork>
    <meminitial>128m</meminitial>
    <maxmem>512m</maxmem>
    <source>1.7</source>
    <target>1.7</target>
  </configuration>
</plugin>
```



# Jar Plugin

- Used to package code into a jar
- <http://maven.apache.org/plugins/maven-jar-plugin/index.html>
- Tied to the package phase
- Configuration section allows customization
  - Includes/Excludes
  - Manifest

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <useDefaultManifestFile>true</useDefaultManifestFile>
      </configuration>
    </plugin>
  </plugins>
</build>
```

# Source Plugin

- Used to attach source code to a jar
- <http://maven.apache.org/plugins/maven-source-plugin/index.html>
- Tied to the package phase
  - Often overridden to a later phase

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-source-plugin</artifactId>
  <version>2.2.1</version>
  <executions>
    <execution>
      <id>attach-sources</id>
      <phase>verify</phase>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



# Javadoc Plugin

- Used to attach Javadocs to a jar
- <http://maven.apache.org/plugins/maven-javadoc-plugin/index.html>
- Tied to the package phase
  - Often overridden to a later phase
- Usually just use the defaults, but many customization options for Javadoc format

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>2.9</version>
  <executions>
    <execution>
      <id>attach-javadocs</id>
      <phase>verify</phase>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

# Summary

- Goals are really just configured plugins in your application
- Plugins are tied to one of the defined phases, but can usually be overridden.
- The compile plugin is already defined for you, but is often changed to use a specific version of Java.
- The jar plugin is one of the default plugins and can be configured to produce artifacts to specific needs.
- Source and Javadocs can easily be generated to be installed in your corporate repository for use by other developers.