

# Developing with Java Build Tools: Ant, Maven, Gradle

Ian Darwin

<http://darwinsys.com/java/>

# *Why Build Tools?*

- External build tools have proven to be the only way to get *repeatable builds*
- Provide consistent set of:
  - Parameters
  - Libraries
  - Tools (compiler etc)
- Used since the 1970's for these reasons

# *Make?*

- The Traditional Unix build tool
- Optimal for C/C++ developers
- No special tools for Java devs
- “JDD” disliked Make's syntax (spaces vs tabs)

# *Ant Buildfile*

- Buildfile contains *targets*
- Target invokes *tasks* to produce a result
- Ant has ~100 predefined tasks
  - Javac, java, jar, war, ...
  - Runs in same JVM as Ant – fast startup

# *Apache Ant*

- JDD created Ant to build Tomcat
  - XML syntax to describe project builds
- People pulled it out to use on other projects<sup>e</sup>
- Became an Apache TLP before Tomcat :-)
- For years was *the* Java build tool

# *Ant Example*

```
<project name="Tools" default="build" basedir=". ">
<property name="src" value="." />
<property name="build" value="." />
<target name="build" depends="init">
    <javac srcdir="{src}" destdir="{build}"
        classpath="{darwin.sys.jar}" />
</target>
<target name="jar" depends="build">
    <jar jarfile="{build}/{deploy.jar}"
        basedir="{build}" includes="META-INF/*,com/**/">
</target>
</project>
```

# *Apache Maven*

- Maven is a higher-level build tool than Ant
- Key benefit: Dependency Management
  - List deps in XML, they download, go on ClassPath
- Focus *what* rather than *how*, e.g., “mvn package”
  - Sensible defaults for where stuff is in the project
- Extensible, e.g., app server vendors have plugins:
  - mvn clean package wildfly:deploy
- Good IDE support

## *Maven can make Maven Projects*

- `mvn archetype:generate` will generate a new project
- Has ~500 templates, or you can specify
- See my `mvndemo-javase`



# *Maven Quick Start*

```
mvn archetype:create \
```

```
-DgroupId=com.darwinsys.demo \
```

```
-DartifactId=mywebproject \
```

```
-DarchetypeArtifactId=maven-archetype-webapp
```

Later: Add a dependency to XML build file (pom.xml)

Eclipsify project: `mvn eclipse:eclipse`

Or: Install m2e from Market, then Configure->Convert to Maven Project

Adds “Maven Classpath Container” to classpath

# *Exploration of some POMs*

- Instructor-led exploration of some existing POMs
- Including Maven-Eclipse integration

# *Where are the jars? Maven Central!*

- Apache project to track huge numbers of projects
- Go to <http://search.maven.org/>, enter keywords, find API
- Copy and paste Maven (or other) dependency into build file.
- Build project. API will download.

# *Gradle*

- The latest kid on the “build tools” block
- Scripting language is based on “groovy” (Java scripting language)
- Has lots of built-in knowledge
- Tends to be more compact, but also more inscrutable
- Chosen to be blessed for Android dev by Google

# *Which to use?*

- Unless there is a reason to do otherwise:
  - Choose Maven
  - Best support, most widely used
- Android Studio users are forced to use Gradle
- Some older projects still use Ant