Introduction to Ant

David Rabinowitz

Ant

- Yet another build tool?
- Why do we need one where there are **make**, **gmake**, **namke**, **imake**, etc. ?

Cross Platform Ability

- A cross platform language needs a cross platform build tool
- Use of XML as the scripting language
  - No more "*!&#$%#ing* tabs"
- Tasks like **javac**, **javadoc**, **mkdir**, **delete**, **jar** and many many more
- Easy to add more tasks
  - **junit**
- Integrated with all major IDEs

Hello world

```
<?xml version="1.0"?>
<project name="Hello World" default="hello">
  <target name="hello">
     <echo message="Hello Ant"/>
  </target>
</project>
```

```
<250|0>dar@sands:~/a> ant
Buildfile: build.xml

hello:
     [echo] Hello Ant

BUILD SUCCESSFUL
Total time: 0 seconds
<251|0>dar@sands:~/a>
```


Concepts

- project
  - default - which is the default target
  - name - the project's name
  - basedir - the base directory from which all path calculations are done.
- target
  - Name - the target's name
  - Depends - a comma-separated list of names of targets on which this target depends.
- Tasks
  - Where the work is really done!

Target Dependencies

- Suppose we have the following:

```
<target name="A"/>
<target name="B" depends="A"/>
<target name="C" depends="B"/>
<target name="D" depends="C,B,A"/>
```

- We call "`ant D`", which of the targets will be the first to be executed?

Core tasks

- Over 80 core tasks!
- We are going to focus on the basic

File System

- Create new directory
  - `<mkdir dir="lib"/>`
- Copy file
  - `<copy file="myfile.txt" tofile="mycopy.txt"/>`
  - `<copy file="myfile.txt" todir="../some/other/dir"/>`
- Zip the contents of a directory
  - `<zip destfile="manual.zip" basedir="htdocs/manual"/>`

File Sets (I)

- How to filter certain files?
  - `<zip destfile="manual.zip" basedir="htdocs/manual" includes="api/**/*.html" excludes="mydocs/**, **/todo.html" />`
- Now the use of `<fileset>` is common
  - `<fileset dir="htdocs/manual"/>`
  - `<fileset dir="." includes="ChangeLog.txt"/>`

File Sets (II)

So we will have:

```
<zip destfile="manual.zip">
  <fileset dir="htdocs/manual"/>
  <fileset dir="."
  includes="ChangeLog.txt"/>
</zip>
```


Properties

- How to define variables or system specific properties?
  - `<property name="dist.dir" value="dist"/>`
  - `<property file="foo.properties"/>`
  - `<property environment="env"/>`
- Use using $ sign:
  - `<zip destfile="${zip.file}" basedir="${docs}/api"/>`
  - `<echo message="Path is ${env.PATH}"/>`

Java (I)

- javac
  - `<javac srcdir="${src}" destdir="${build}" classpath="xyz.jar" />`
  - `<javac srcdir="${src}:${src2}" destdir="${build}" includes="mypackage/p1/**,mypackage/p2/**" excludes="mypackage/p1/testpackage/**" classpath="xyz.jar" debug="on" />`

Java (II)

- jar
  - `<jar destfile="${dist}/lib/app.jar" basedir="${build}/classes"/>`
  - `<jar destfile="${dist}/lib/app.jar" basedir="${build}/classes" excludes="**/*Test.class" />`

Java (III)

- junit

```
<junit>
  <test name="my.test.TestCase"/>
</junit>
```


Java (IV)

- Execute Java files

```
<target name="transform" depends="compile">
  <java classname="
  org.codehaus.aspectwerkz.compiler.AspectWerkzC"
  fork="true">
      <jvmarg value="-Daspectwerkz.definition.file=
                ${src.dir}/aspectwerkz.xml"/>
      <arg value="${build.dir}"/>
  </java>
</target>
```


More tasks

- tar, untar, gzip, gunzip
- move, delete
- CVS
- tstamp
- mail
- And many many more…

Defining new tasks

- taskdef - Adds a task definition to the current project
  - `<taskdef name="myjavadoc" classname="com.mydomain.JavadocTask" />`
- The task must extend `org.apache.tools.ant.Task`
  - How does ant knows how to set the task's attributes?

Conclusion

- The *de-facto* build tool of the Java world
- Cross-platform
- Target dependencies
- http://ant.apache.org/manual/