

Introduction to Ant- a framework example

Amit Shabtay

Frameworks

- "A set of cooperating abstract and concrete classes that makes up a reusable design for a specific class of software"
- An Object-Oriented Reuse Technique
 - Design Reuse + Code Reuse

March 3rd, 2004

Object Oriented Design Course

2

Designing an OO Framework

- Domain Knowledge
 - What applications is the framework for?
 - What is common to all of them?
- Architecture
 - Biggest, most critical technical decisions
 - What is required besides classes?
- Object-oriented design
 - Design Reuse: Patterns
 - Inversion of Control + Find right hooks

March 3rd, 2004

Object Oriented Design Course

3

Ant



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

March 3rd, 2004

Object Oriented Design Course

4

Ant- 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

March 3rd, 2004

Object Oriented Design Course

5

Hello world

```
<?xml version="1.0"?>
<project name="Hello World" default="hello">
  <target name="hello" description="Prints hello Ant">
    <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>
```

March 3rd, 2004

Object Oriented Design Course

6

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!

March 3rd, 2004

Object Oriented Design Course

7

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?
- Which of them will probably be the default?

```
<project name="proj" default="???">
```

March 3rd, 2004

Object Oriented Design Course

8

Examples

- What happens here?

```
<target name="A" />
<target name="B" />
<target name="*" />
```

* is called a wild card
- And here?

```
<target name="build-module-A"
if="${module-A-present}" />
```

March 3rd, 2004

Object Oriented Design Course

9

Core tasks

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

March 3rd, 2004

Object Oriented Design Course

10

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" />`

March 3rd, 2004

Object Oriented Design Course

11

File Sets (I)

- How to filter certain files?
 - `<zip destfile="${dist}/manual.zip" basedir="htdocs/manual" includes="api/**/*.html" excludes="**/todo.html" />`
- The use of `<fileset>` is common
 - `<fileset dir="htdocs/manual" />`
`<fileset dir="." />`
`includes="ChangeLog.txt" />`

March 3rd, 2004

Object Oriented Design Course

12

File Sets (II)

So we will have:

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

March 3rd, 2004

Object Oriented Design Course

13

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}"/>
(PATH should be Path on windows)

March 3rd, 2004

Object Oriented Design Course

14

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" />

March 3rd, 2004

Object Oriented Design Course

15

Java (II)

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

March 3rd, 2004

Object Oriented Design Course

16

Java (III)

- junit

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

March 3rd, 2004

Object Oriented Design Course

17

Java (IV)

- Execute Java files

```
<exec executable="emacs">
  <env key="DISPLAY" value=":1.0"/>
</exec>

<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>
```

March 3rd, 2004

Object Oriented Design Course

18

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 know 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/>