Apache Ant Java build tool

CONCEPTS

Ant reads build.xml in the current directory. Run with no target and it runs the project's default target.

Project build.xml, default target
- **Target** named step, depends on others
  - **Task** javac, jar, copy ...

Properties are immutable: the first definition wins, so -D on the command line overrides build.xml.

depends only orders targets. It does not rerun a target that already ran in this invocation.

RUN

ant
Run the project's default target

ant *<target>* **[***<target>***...]**
Run targets in order

ant -projecthelp / ant -p
List targets that have a description

ant -buildfile *<file>* **/ ant -f** *<file>*
Use a different build file

RUN (cont.)

ant -find [*<file>***] / ant -s [***<file>***]**
Search parent dirs for the build file

ant -version
Print Ant version

ant -diagnostics
Print Java, classpath and Ant diagnostics

ant -help / ant -h
Show command line options

ant -keep-going / ant -k
Continue with targets not depending on a failure

ant -noinput
Never prompt for input (CI)

PROPERTIES AND OUTPUT

ant -D*<name>***=***<value>*
Set a property from the command line

ant -propertyfile *<file>*
Load properties from a file

ant -quiet / ant -q
Less output

ant -silent / ant -S
Only task output and failures

ant -verbose / ant -v
Extra output

ant -debug / ant -d
Debugging output

PROPERTIES AND OUTPUT (cont.)

ant -logfile *<file>* **/ ant -l** *<file>*
Write the log to a file

ant -lib *<path>*
Add jars or dirs to Ant's classpath

ANT_OPTS="-Xmx1g" ant
Pass options to the JVM

BUILD FILE BASICS

<project name="<n>" **default="***<target>***" basedir=".">**
Root element

<target name="<n>" **depends="***<a>***,***<b>***" description="***<text>***">**
Define a target

<property name="<n>" **value="***<v>***"/>**
Define a property

<property file="build.properties"/>
Load properties from a file

<property environment="env"/>
Read env vars as ${env.NAME}

${*<name>***}**
Reference a property

${basedir}
Project base directory

BUILD FILE BASICS (cont.)

<target name="<n>" **if="***<prop>***">**
Run only if a property is set

<target name="<n>" **unless="***<prop>***">**
Run only if a property is not set

<import file="<other.xml>"/>
Include another build file

<path id="<id>"><fileset dir="lib" includes=".jar"/></path>*
Reusable classpath

<fileset dir="<d>" **includes="**/*.java" excludes="***<pat>***"/>**
Select files by pattern

<echo message="<text>"/>
Print a message

<exec executable="<cmd>"><arg value="<a>"/></exec>
Run an external command

COMMON TASKS

<mkdir dir="<d>"/>
Create a directory

<delete dir="<d>"/>
Remove a directory tree

<copy todir="<d>"><fileset dir="<s>"/></copy>
Copy files

COMMON TASKS (cont.)

<move file="<f>" **tofile="***<t>***"/>**
Move or rename a file

<javac srcdir="src" destdir="build/classes" classpathref="<id>" **includeantruntime="false"/>**
Compile Java sources

<jar destfile="dist/<n>.jar" **basedir="build/classes"/>**
Build a jar

<manifest><attribute name="Main-Class" value="<cls>"/></manifest>
Set jar manifest (inside <jar>)

<java classname="<cls>" **classpathref="***<id>***" fork="true"/>**
Run a Java class

<junit printsummary="yes" haltonfailure="yes">
Run JUnit tests

RECIPES

Clean build with a property

```
ant -p
ant clean jar -Dversion=1.2
# in build.xml: <jar
destfile="dist/app-
${version}.jar" .../>
```