

COMP220/285 Lab sessions 4-5

Contents

Introduction	1
Ant: Structured Build	1
Ant: Working with filesets.....	2
Ant: Working with properties	3

Introduction

The point is that working jointly with Eclipse and Ant without experience may be overcomplicated and may serve as a source of possible mistakes.

Ant: Structured Build

Please, follow slides Ant: Structured Build (Lecture 7)

Start with creating the directory

M:\comp220_285\build2

which will serve in this Lab as the current directory or base directory.

Create also subdirectories src and build\classes according to the slide 6.

Continue reading these slides making step-by-step all actions required there. These actions are typically labelled by the word TRY or RUN or in other way by using red colour. Do not hurry up! Achieve the full understanding of all the comments in slides.

This may take a longer time. Please finish it yourself. This way you will get a good understanding of the initial concepts of Ant.

Optional material (Self-Study). At the end of the above portion of work (see Slides 43-46) you will need the extended file build.xml. Save it in your build2 directory (Right-Click -> Save-Link-As...; It is a wrong idea to open an XML file by a browser and then copy-and-paste it into an editor window!).

Ant: Working with filesets

Please, follow slides Datatypes and Properties and Filesets

First, read and understand Slides 1-20 and do exercises from Slides 20-22. You will continue working in

M:\comp220_285\build2

possibly creating appropriate new files and subdirectories.

Please do not change already existing files for future revision of your work. You only can delete directories build and dist which are easily recoverable with their content by running in the above directory the command

ant -f structured.xml

It makes sense to create one build file (call it, say, filesets.xml) to work in it with various filesets. Additionally, experiment with a bit more complicated filesets having several <include> and <exclude> elements

In fact, it is assumed that you will be creative enough to do these simple exercises. For example, you could generate several directories and subdirectories and some artificial files in them with various extensions (just create a text file and change its extension in any way you need) to experiment with <include> and <exclude> elements.

Additionally, create in your build file filesets.xml a complicated fileset with an id, and then copy this fileset by using the attribute refid to simplify the form of the <copy> task.

You can invent your own experiments to check that Ant really does things described in slides considered in lectures.

Please, follow slides Datatypes and Properties and the following instructions.

1. Built-in properties

Create new build file

M:\comp220_285\build2\builtinprop.xml

which will serve to check various built-in properties of Ant and JVM system properties from Slides 4-9. Check also the behaviour of basedir depending on how it is declared in this build file or in command line (Slides 7,8).

2. Setting and using properties by the tasks `<property>` and `<available>` (and also `<uptodate>` and `<condition>` - to be considered later).

Just read Slides 10-16 to get general understanding. Then do the following actions.

Setting and using a simple property (Slide 11). You can check what is the value of a property by appropriate using `<echo>` task. Another way to see the difference of two values on and off of build.debug property is to compare (precise) sizes of compiled files according to a brief explanation on (Slide 11).

Loading properties from a property file; check with Ant the values of properties build.debug and output.dir (Slide 12).

Fixing properties to absolute path locations. Check what is the difference between value and location attribute of the `<property>` task (Slide 14).

Check immutability of properties (Slides 15,16).

Experiment with the task `<available>` according to Slides 26-32. It makes sense to do these exercises in a specially created an Ant file with the same name

.

• H:\Antbook\ch02\secondbuild\available.xml

(with several appropriate targets) in the old directory which already contains the necessary subdirectories and files there.

Important experiment with JAR files

When running structured.xml build file, you created project.jar file. Rename its extension .jar with .zip and click on this file to see what is inside. Which files from build/classes are archived there? Under which system of subdirectories? Compare this with the content of the archived directory build/classes. (For simplicity, we ignore META-INF in this comparison.)