



build tools

CSC532

nrg



Make utility

- software engineering tool for managing and maintaining a software construction
- Beneficial for complex software
- a descriptor file “makefile” containing dependency rules including target and actions (commands)
- Reduce compile time by checking dependency and timestamp



Materials here are excerpted from Intro to Unix Make, <http://www.mtsu.edu/~csdept/FacilitiesAndResources/make.htm>

How Make works

- Make goes through a descriptor file
- Start with the main target.
- Make looks at each of the target's dependencies.
- Follows the chain of dependencies until it reaches the end of the chain.



Simple example

- Say, you want to build a prog1 from 3 C++ files.

```
% CC -o prog1 file1.cc file2.cc file3.cc
```

- If you modify one of them, and recompile prog1, every single cc files will be compiled again.



Simple example of Makefile(continued)

```
prog1 : file1.o file2.o file3.o
    CC -o prog1 file1.o file2.o file3.o
file1.o : file1.cc mydefs.h
    CC -c file1.cc
file2.o : file2.cc mydefs.h
    CC -c file2.cc
file3.o : file3.cc
    CC -c file3.cc
clean :
    rm file1.o file2.o file3.o
```



Compile your program with make

- If you name your make description file as “makefile”, you can simply issue “make”

% make

- If you want to explicitly specify your description file, you can use **-f** option

%make -f makefile



<ant>: XML build tool



- Open source from Jakarta.
- Similar build tools like *make, gnumake, nmake*.
- Implemented in Java, and O.S. independent
- Configuration files are XML based.

- REQUIREMENTS

 - JAXP Parser (*xerces.jar*)

 - JDK 1.2 or higher

 - ANT_PATH



- Every project using ant has a default *build.xml*

- Inside *build.xml*

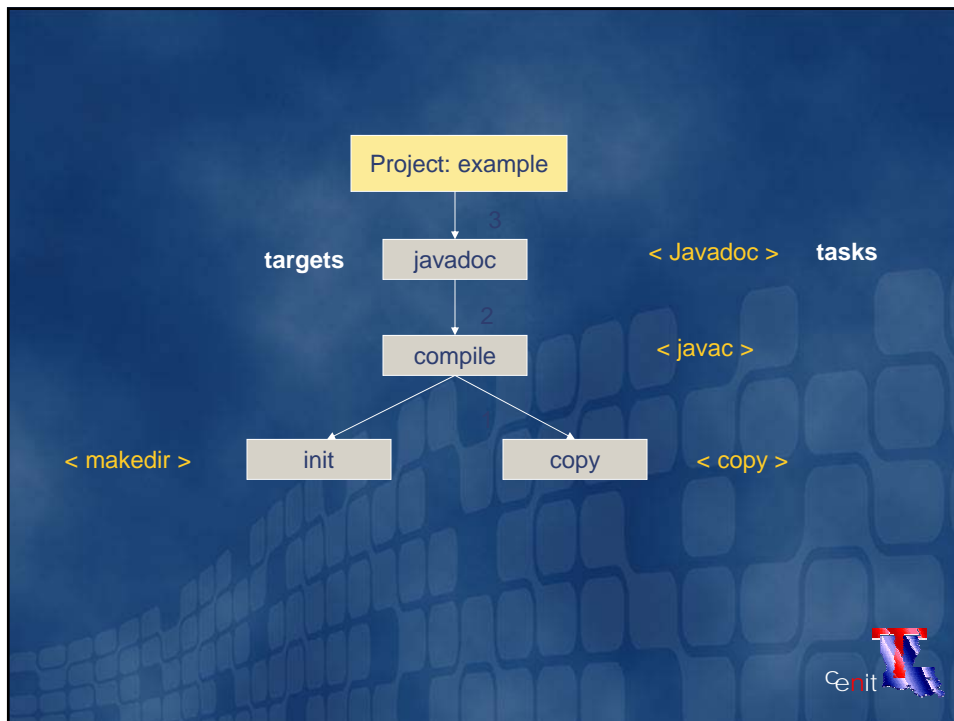
-- Each build file contains one *project* and a default *target*

ex: Compile is a target

-- *targets* contain *task* elements

ex: Tasks for compile, are copy, compile, delete

-- *tasks* have values assigned to attributes



SAMPLE BUILD FILE

```
<?xml version="1.0"?>
<project name="example" default="javadoc"
  basedir=".">

  <property name="name" value="example" />
  <property name="source" value="source" />
  <property name="build" value="${basedir}/build"
  />
  <property name="docDir" value="doc" />

  <target name="init" >
    <mkdir dir="${build}" />
    <mkdir dir="${source}" />
    <mkdir dir="${docDir}" />
  </target>
```



```
<target name="copy">
  <copy todir="${source}">
    <fileset dir="${basedir}" includes="*.java"/>
  </copy>
</target>

<target name="compile" depends="init,copy" >
  <javac srcdir="${source}" destdir="${build}" >
  </javac>
</target>

<target name="javadoc" depends="compile">
  <javadoc sourcepath="${source}"
  packagenames="Adder.*"
  destdir="${docDir}" />
</target>

</project>
```



- Output indicates the tasks that are executed like
init, copy, compile
- Can build your own tasks
org.apache.tools.ant.Task
- Gui version of ant is still under development



< end >

