Computer Science 507
Software Engineering
The College of Saint Rose
Spring 2013

# Lab 2: Build Management

**Due: TBD**

In this assignment, you will learn or refersh your knowledge of the `make` and Ant programs that can be used to manage project builds. You may work alone or with a partner on this assignment.

---

## Getting Set Up

Create a directory in your account on `mogul.strose.edu` for your work on this lab (a directory `lab2` inside the `cs507` directory you created for the first lab might be appropriate).

In your favorite editor, create a document in which you will answer the questions you find in this lab. Start by putting your name at the top of this document.

---

## Using the `make` Utility

Any non-trivial software development involves many iterations of editing, compiling, linking, and running your programs. The code will be spread across multiple files. The most common mechanism for managing this process when programming in C in a Unix environment is the `make` utility. Its utility is not restricted to C programs. The actions of `make` are specified by rules in a `Makefile`. Copy the following example to your account on `mogul.strose.edu`:

**See Example:**
`/home/cs507/examples/make-example`

You should find a small C program that demonstrates the use of multiple source files and a `Makefile`. Compile the program with `make`.

**Question 1:** What is the output produced when you run `make`? (2 points)

Now, look at the rules and the description in the `Makefile`.

**Question 2:** Briefly describe how `make` uses the rules in the `Makefile` to produce the executable `main`. Be sure to include the series of targets, their dependencies, and the commands used to satisfy those dependencies for each target. (7 points)

---

## Apache Ant for Java Build Management

While `make` can be used to manage builds of a Java project, many Java projects use a tool called Apache Ant (`http://ant.apache.org/`. Read the first section on that page to get a brief description of Ant.

Our goal today is only to see the basics of Ant, so we will use an example that can build a “Hello,

World" Java program using Ant.

- Start at `http://passion4java.blogspot.com/2009/10/hello-world-example-for-a` `html`.
- In your subdirectory for this lab, create another subdirectory named `ant`. Inside of `ant`, create subdirectory named `src`, and in that directory, place the Java program shown in blue near the top of the page as the file `AntHelloWorld.java` (replacing the printout with a message of your own).
- Then, in your `ant` subdirectory, create a file `build.xml` which contains the red text under "Hello world build script".
- Still in your `ant` directory, issue the command `ant`.

  **Question 3:** What output is produced? (2 points)
- Run the program with the command

```
java -cp AntHelloWorld.jar com/passion4java/helloworld/ant/AntHelloWorl
```

  **Question 4:** What output is produced? (2 points)

You have now compiled a Java program using Ant and run that Java program from a jar (Java ARchive) file.

**Question 5:** Paste in the output of the command `ls -lR` from inside your `ant` directory. (2 points)

---

## Submission and Grading

Transfer the files you have created for this lab to the computer you're working on with a secure copy program like WinSCP or FileZilla.

To submit the assignment, send your answers to the questions to *terescoj@strose.edu* by TBD.

Please include a meaningful subject line (something like "CS507 Lab 2 Submission"). Please do not include any additional files, such as emacs backup files, object files, or executable programs.

This lab will be graded out of 15 points according to the breakdown shown by the lab questions above.