CSE 331 Software Design and Implementation

Handout T5: *Working at Home*

Contents:

- The Problem
- Step 1: Get Oracle's Java Development Kit (JDK)
- Step 2: Setup the JAVA_HOME Environment Variable
  - Windows
  - Linux
- Step 3: Get Eclipse
  - Where to Put Eclipse on Your Machine
  - Setting the JDK in Eclipse on Windows
- Step 4: Use SVN
  - Eclipse
    - Install Subclipse SVN Plugin
    - Checkout an assignment
  - Command Line
- Step 5: Turn in an assignment
- Logging in to attu with SSH
  - SSH on Linux
  - SSH on Windows
  - Server Authentication Warning
  - File Transfer: SCP
- **Disclaimers About Using Your Own Tools**

The Problem

Many students prefer to work at home than in the UW CSE instructional labs. But the nice thing about a UW CSE machine is that it already has all of the necessary software for CSE 331 installed and is properly configured whereas your computer at home probably is not. We strongly suggest using the CSE lab machines (such as attu), and we will provide support if you have problems with your account. Failing that, you can use the CSE-provided instructional home VM, which nearly replicates the attu environment, and we will also provide support if you have problems with an un-customized home VM. If you wish to set up your own machine to permit you to do your CSE 331 work directly rather than in a virtual machine, this document tells you how to do so. However, the CSE 331 staff will not provide support (beyond these instructions) for setting up the CSE 331 tools on your own machine; if you have trouble, you are on your own.

Note that even if you choose to work at home, you will still need to complete the Initial Setup on `attu`.

Step 1: Get Oracle's Java Development Kit (JDK)

You want to download Oracle's **J2SE v7.0 JDK** from their web site. This is somewhat tricky because there are a lot of things on the download page with nearly the same name. For example, the **JRE** is the **Java Runtime Environment** which lets you run Java programs, but does not provide the tools for Java development.

Thus, go to http://www.oracle.com/technetwork/java/javase/downloads/ and make sure that you follow the link labeled **Download JDK**.

Step 2: Setup the JAVA_HOME Environment Variable

Create an environment variable called **JAVA_HOME** that points to the directory in which you installed the JDK. To get javadoc to work correctly, you also need to add the **bin** directory of **JAVA_HOME** to your **PATH** environment variable.

Windows

To set environment variables on Windows, select **System** from **Control Panel**, go to the **Advanced** tab and hit the **Environment Variables** button. Add a new variable to the **System variables** called **JAVA_HOME**. Its value will be the location of the SDK (which is most likely **C:\Program Files\Java\jdk1.7.0_21**). Then append **%JAVA_HOME%\bin** to your **PATH** environment variable under **System variables**. This will put the executables associated with the JDK in your path. On Windows, PATH elements are separated by semicolons, so you may have to add a semicolon to the end of your **PATH** variable before appending **%JAVA_HOME%\bin**.

Linux

To set environment variables in bash, add the following lines to your `.bashrc` file:

```
export JAVA_HOME wherever you installed the JDK
export PATH=$PATH:$JAVA_HOME/bin
```

Step 3: Get Eclipse

In CSE 331, we will use **version 4.2 (Juno)** of Eclipse.

Where to Put Eclipse on Your Machine

Eclipse does not come with an installer, so this confuses many people. Basically, you download it, unzip it into a directory, and then run the executable in that directory to start Eclipse. There are some small bugs in Ant within Eclipse that manifest themselves if it is installed in a directory with spaces in its name, so instead of installing it into a directory such as:

```
C:\Documents and Settings\Joe User\Desktop\eclipse
```

or

```
C:\Program Files\eclipse
```

use:

```
C:\eclipse
```

Setting the JDK in Eclipse on Windows

If you are using a Windows, setup your Eclipse workspace to use the JDK as described in the Tools: Editing, Compiling, Running, and Testing Java Programs handout.

Step 4: Use SVN

Eclipse

Install Subclipse SVN Plugin

In CSE 331 we use the Subclipse SVN Plugin for Eclipse to manage the assignments. Follow the official installation instructions here to install the Subclipse SVN Eclipse plugin, as well as the special directions for Windows and Mac OSX users below.

Windows users will need to download a SSH client (if they haven't already) before the Subclipse Plugin. SSH client options include TortoiseSVN or PuTTY. After you install one of these, install Subclipse and continue the rest of the set up process. This is done to prevent a common Subclipse error: "SVN Cannot Create Tunnel." For additional information refer to this blog post.

In the past, Mac OSX users have needed to install Subclipse 1.6.x rather than 1.8.x (the latest version at the time of this writing) to match the latest available version of JavaHL.

If you have trouble using Subclipse, try installing SVNKit and configuring Subclipse to use that instead of JavaHL. This has solved problems for some students.

Subclipse is a handy tool: it has nice features and is integrated into your development environment. However, it is also buggy. If you have trouble with it, don't get frustrated or stuck. The command-line tool `svn` tends to be more reliable, and you should feel comfortable using it as well. It works on all operating systems. Another option, which is also more reliable than Subclipse, is TortoiseSVN, which only works on Windows computers.

Checkout an assignment

When the first assignment has been released, you can check out the `cse331` project from your SVN repository by following the checkout instructions for Windows.

Command Line

The command line should function the same as in the Allen Center software labs, with the exception that when you checkout, you must specify the repository URL as `svn+ssh://attu.cs.washington.edu/projects/instr/13sp/cse331/`*`YourUserName`*`/REPOS`. For example, to checkout the `cse331` directory you should run:

```
svn checkout
svn+ssh://YourUserName@attu.cs.washington.edu/projects/instr/13sp/cse331/YourUserName
/REPOS/cse331 cse331
```

Step 5: Turn in an assignment

Once you are done with an assignment, you must commit it to SVN, because we collect the assignment by retrieving whatever is in your SVN repository at the deadline. (However, it is recommended to commit frequently, not just when you have completed the assignment.) Also, it is *highly recommended* to validate the assignment on `attu`.

a. In the Package Explorer window, right-click on "cse331" in the directory structure (the one representing the entire project). Choose **Team >> Commit...**
b. On `attu`, first make sure you have followed the checkout instructions to get a `cse331` directory. Then, validate your assignment.

Logging in to attu with SSH

`attu` is the name of an Instructional Workstation (IWS) Linux machine. You will occasionally need to log into this machine. How you log into attu depends on whether you are starting from a Linux machine or a Windows machine.

SSH on Linux

a. Obtain a command prompt (a.k.a. a terminal or console). You can do so by clicking on Applications -> System Tools -> Terminal.
b. At the command prompt, run the following command:

```
ssh YourCSENetID@attu.cs.washington.edu
```

Use the same password you use to login to the Linux machines in the Allen Center software labs.

NOTE (for those who are new to the command line): When you try to type passwords in the command line, you may be alarmed that you can't see any text entered. Don't Panic. In order to protect your password your typing simply isn't being shown. Just type your password as normal and press enter.

If you are using the Linux machines in an Allen Center software lab, you can alternatively use a shorter version of the command:

```
ssh attu
```

This works because the username defaults to the username you are currently logged in with and the target domain defaults to the domain of the machine you are connecting from.

However, you should not need to log into `attu` from an Allen Center Linux IWS machine! Such a machine already has access to the file system based on your CSE account, and `attu` will be no different.

SSH on Windows

- From a CSE instructional machine: Double-click on the **SSH attu** shortcut on the Desktop.
  If the shortcut is not on your Desktop, go to: **All Programs » UNIX Connectivity » SSH » SSH attu** in the start menu.
- From home:
  a. Install an SSH client. A great (and free) one is PuTTY.

b. Using PuTTY, connect to `attu.cs.washington.edu` by entering it in the `Host Name` textbox and clicking `Open`. You will be prompted for your username and password.

In either case, your username is your **CSENetID**, and your password is the same one you use to login to the *Linux* machines in the Allen Center software labs.

Server Authentication Warning

The first time you connecting to `attu` from a given machine, you will receive a server authenticity warning like this:

```
The authenticity of host 'attu.cs.washington.edu (128.208.1.139)' can't be established.
```

Along with the warning, the SSH client will display the RSA key fingerprint of the remote host so that you can verify the host's identity (verifying helps to prevent man-in-the-middle attacks). In our case, it is safe to say "yes" to continue connecting. When you connect, SSH will cache the host key in order to automatically verify the remote host's identity in the future.

A good article about SSH host keys can be found at http://www.symantec.com/connect/articles/ssh-host-key-protection.

File Transfer: SCP

If you only want to transfer files between your CSE account and your home machine, you can use `scp` ("secure copy"). **For CSE 331, you should rarely, if ever, need to manually transfer files.** All your code and other homework materials will be in a Subversion repository, allowing you to automatically and safely synchronize your work across machines. We provide information about `scp` as a reference for your general knowledge.

On a Mac or Linux machine, you can run scp at the command line. Run `man scp` for documentation on this command. If you prefer a graphical interface or if you are using Windows, you can install a file transfer program such as WinSCP. Most of the directions above for establishing a remote connection via ssh also apply to scp.

Disclaimers About Using Your Own Tools

All code that you turn in must run on `attu` using the staff-supported version of Java, which is Java 7. The CSE 331 staff provides technical support only for the staff-supported version, and resolving any discrepancies between Java versions is your own responsibility.

You may use whatever tools you like to develop your code. The CSE 331 staff provides technical support only for using Eclipse at UW CSE. This is not because we think that this setup is the only way (or even the best way) to develop Java code, but because we cannot be expected to be familiar with every operating system or text editor that you might choose to use.

If you choose to use Eclipse on your own machine instead of on a UW CSE workstation, and you have problems, then you are generally on your own (The CSE 331 staff will try to help you, but only if you are running either Linux or Windows 7 or Vista, and we make *no guarantees*.) That being said, it is generally safe to expect Eclipse to behave the same way on all operating systems on which it is supported (which currently includes: Windows 98 and later, Mac OSX, Linux, Solaris 8, AIX, and HP-UX). We know that the UW CSE workstations are not the most powerful machines in existence, so you may find that Eclipse works better when you run it from your own computer.

Back to the CSE 331 home page.
For problems or questions regarding this page, contact: cse331-staff@cs.washington.edu.