6.170 Laboratory in Software Engineering
Working at Home

Contents:

Step 0: The Problem

Many students prefer to work at home than in Athena. But the nice thing about an Athena machine is that it already has all of the necessary software for 6.170 installed and is properly configured whereas your computer at home probably is not. Although the 6.170 staff will not personally provide support for setting up the 6.170 tools on your own machine, we provide these instructions to help you set up the 6.170 tools on your own machine, if you like.

Note that even if you choose to work at home, you will still need to run

```
add 6.170
student-setup.pl
```

once on Athena.

Step 1: Get Sun's Java Software Development Kit (SDK)

You want to download Sun's **J2SE v5.0 SDK** 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. Currently, J2SE v5.0 is not available for Apple OSX.

Thus, go to http://java.sun.com/j2se/1.5.0/download.html and make sure that you follow the link labeled **Download J2SE SDK**. Instead of downloading **JDK 5.0** bundled with Netbeans (since you'd be using Eclipse), download the **JDK 5.0 Update 1** package. (You may also find it useful to download the **J2SE 1.5.0 Documentation**, as it will faster to access the documentation from your local machine than it is to access it over the web.)

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 SDK. To get javadoc to work correctly, you also need to add the **bin** directory of **JAVA_HOME** to your **PATH** environment variable.

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:\j2sdk1.5.0_01**). Then append **%JAVA_HOME%\bin** to your **PATH** environment variable under **System variables**. This will put the executables associated with the SDK 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**.

To set environment variables on tcsh (the default shell on Athena), add the following lines to your `.cshrc` (or `.cshrc.mine`) file:

```
setenv JAVA_HOME wherever you installed the sdk
setenv PATH ${PATH}:${JAVA_HOME}/bin
```

To set environment variables in bash (another type of Linix shell), add the following lines to your `.bashrc` file:

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

Step 3: Get Eclipse

To download Eclipse, go to http://www.eclipse.org/downloads/index.php and select your operating system. You will be using Eclipse version 3.1 for this course because it is the only known version of Eclipse that supports all the Java 1.5 semantics.

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 direcory such as:
`C:\Documents and Settings\Joe User\Desktop\eclipse`
or
`C:\Program Files\eclipse`
use:
`C:\eclipse`

Step 4: Use CVS

Set up CVS in Eclipse

You will have to make the following changes to your Team preferences in Eclipse 3.1:

1. Go to **Window** >> **Preferences** >> **Team** >> **CVS** >> **Ext Connection Method**
2. Select **User another connection method type to connect**
3. For **Connection Type** select **extssh**

Checkout lib6170

We have created a CVS module called **lib6170**. This module contains all the libraries that you need to compile the code for this and future problem sets. You only have to check this out once.

1. Select **Window** >> **Open Perspective** >> **Other...** >> **CVS Repository Exploring**
2. Right-click in the `CVS Repositories' Window, and select **New** >> **Repository Location...**
3. Fill in the fields as follows, and as shown below:
4. Host: **athena.dialup.mit.edu**
5. Repository path: **/mit/6.170/lib-cvsroot**
6. User: *<your username>*
7. Password: <your password>
8. Connection type: **extssh**
9. Click **Finish**
10. Expand the CVS Repository directory structure, in the CVS Repositories Perspective Window. Expand **HEAD**, so that "lib6170" is visible.
11. Right-click "lib6170", and select **Check Out As...**

Checkout a Problem Set

If you have already run `student-setup.pl` on Athena, you are now ready to checkout your first problem set.

1. Select **Window** >> **Open Perspective** >> **Other...** >> **CVS Repository Exploring**
2. Right-click in the `CVS Repositories' Window, and select **New** >> **Repository Location...**
3. Fill in the fields as follows, and as shown below:
4. `Host:` **`athena.dialup.mit.edu`**
5. `Repository path:` **`/mit/<your username>/6.170/cvsroot`**
6. `User:` *`<your username>`*
7. `Password: <your password>`
8. `Connection type:` **`extssh`**
9. Click **Finish**
10. Expand the CVS Repository directory structure, in the CVS Repositories Perspective Window. Expand **HEAD**, so that "psN" is visible.
11. Right-click "psN", and select **Check Out As...**

[**Note:** you will only need to do steps 1-4 once and repeat steps 5 and 6 for each problem set you want to checkout]

Turn in a Problem Set

Once you are done with a problem set, you will need to commit it to CVS and then validate it on Athena.

1. In the Package Explorer window, right-click on the top-most "psN" in the directory structure (the one representing the entire project). Choose **Team** >> **Commit...**
2. On Athena run the commands described in the "Turn In Your Problem Set" step of Problem Set Procedure in the assignments section.

6.170 Disclaimers About Using Your Own Tools

For 6.170, we only officially provide technical support for using Eclipse on Athena. 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. Hence, the following disclaimers:

1. If you choose to use an IDE other than Eclipse, and you have problems, then you are on your own. If you should decide to use Emacs, TAs and LAs will be able to help you out, but you are strongly encouraged to use an IDE. Using an IDE has

many advantages over an ordinary text editor, so even if you choose not to use Eclipse, you may want to try out one of the other IDEs, such as IntelliJ or NetBeans, before opting to use a text editor, such as Emacs or Textpad.

2. If you choose to use Eclipse on your own machine instead of on an Athena workstation, and you have problems, then you are again generally on your own, unless you are running either Linux or Windows XP. The 6.170 staff should be able to help you with Eclipse installation on these platforms, though we cannot be expected to serve as tech support for any other operating system. 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 Athena workstations are not the most powerful machines in existence, so you may find that Eclipse works much better when you run it from your own computer.