

6.170 Laboratory in Software Engineering

Emacs and Command Line Reference for 6.170

This document is meant to be a quick reference for Emacs and Command Line commands. If you want to know more about what each command does, take a look at the Problem Set Procedure document or one of the other Tools documents.

Contents:

- [CVS](#)
 - [Setting up CVS in Your Environment](#)
 - [Checkout the Problem Set from CVS](#)
 - [How Do I Add a File to CVS?](#)
 - [Committing Changes to CVS](#)
 - [Updating Your Local Copy of the CVS Repository](#)
- [Compiling Code and Running Programs](#)
 - [Compiling Code](#)
 - [Running your program from the Command Line](#)
 - [Running your program from Emacs](#)
 - [Running JUnit Tests](#)
- [Editing Code in Emacs](#)
 - [Creating a New File](#)

CVS

Setting up CVS in Your Environment

Setting up CVS in Athena	Setting up CVS from Home
<p>student-setup.pl should have set up CVS for you. To make sure, check to see that the <code>CVS_RSH</code> environment variable is set to <code>ssh</code> and the <code>CVSROOT</code> environment variable is set to <code>/mit/<your username>/6.170/cvsroot/</code> by typing</p> <pre>echo \$CVSROOT echo \$CVS_RSH</pre> <p>to print out the values of those variables.</p>	<p>Similarly, you're going to want to set your <code>CVSROOT</code> environment variable to <code>athena.dialup.mit.edu:/mit/<your username>/6.170/cvsroot</code> and set your <code>CVS_RSH</code> variable to <code>ssh</code>. Check this by typing</p> <pre>echo \$CVSROOT echo \$CVS_RSH</pre> <p>to print the current setting of these variables. If they're not set correctly, take a look at the <code>export</code> and <code>setenv</code> commands, depending on whether your shell is <code>bash</code> or <code>tcsh</code>.</p>

If you are not comfortable with using `vi` (or if you have never heard of `vi`), you may want to set your default editor to Emacs. To do this, first run:

```
echo $SHELL
```

If your shell is `/bin/athena/tcsh`, you can set your default editor to emacs by running:

```
setenv EDITOR emacs
```

If your shell is `/bin/athena/bash`, you can set your default editor to emacs by running:

```
export EDITOR=emacs
```

Checkout the Problem Set from CVS

How to Checkout from the Command Line	How to Checkout in Emacs
<ol style="list-style-type: none"> 1. <code>cd ~/6.170</code> 2. <code>cvs checkout psN</code> 	<ol style="list-style-type: none"> 1. In emacs, type <code>Alt-x cvs-checkout</code> 2. Fill out the following information when prompted: Module: <code>psN</code> CVS Checkout Directory: <code>~/6.170</code>

`psN` is the name of the **module** that you are checking out from CVS. Checking out will create the directory `~/6.170/psN/` which will have the contents of your problem set.

How Do I Add a File to CVS?

Adding a File to CVS from the Command Line	Adding a File to CVS in Emacs
<p>When adding a file, you must indicate to CVS whether it is a text/ASCII file (such as a <code>.java</code> file) or a binary file (such as a <code>.jpg</code> image) when invoking the <code>cvs add filename</code> command. If you're already in your <code>~/6.170/psN/</code> directory, you can add a text/ASCII file as follows:</p> <pre>cvs add src/psN/RandomHello.java</pre> <p>But if the file is binary, then you need to use a <code>-kb</code> flag:</p> <pre>cvs add -kb doc/screenshot.jpg</pre>	<ol style="list-style-type: none"> 1. Switch Emacs into CVS mode by running <code>Alt-x cvs-examine</code> or <code>Alt-x cvs-update</code>. When prompted for CVS Examine (directory), enter <code>~/6.170/psN/</code>. This will bring up a new buffer that lists all the new and modified files in your working directory. 2. In this CVS buffer, you can move the keyboard cursor over a file and issue one of the following commands: <ul style="list-style-type: none"> o Pressing 'a' will add an "Unknown" file into the repository. Remember that you must then commit this file. o Pressing 'c' will commit a "Modified" or "Added" file. o Pressing 'r' will remove the

	current file from the repository. Again, you must make sure you commit this file.
--	---

Some of these commands will prompt you (either in the minibuffer or a new window) to enter a short description or log message. Keep in mind that entering a good log message now can save you time later if you need to revert back to an older version.

Committing Changes to CVS

Committing a File from the Command Line	Committing a File in Emacs
<p>1. You can commit a group of <i>filename(s)</i> to CVS by running:</p> <pre> cvs commit -m "a descriptive log message" filename(s) </pre> <p>If you omit the message flag from your <code>commit</code> command, CVS will throw you into an editor where you can enter a message. If you have not set your <code>\$EDITOR</code> environment variable to <code>emacs</code>, then this editor will most likely be <code>vi</code>. If you have never used <code>vi</code> before, you probably won't be able to figure out how to exit from it. The simplest thing to do is to type <code>:wq</code> to write-quit, cancel your <code>cvs commit</code> command, and set your <code>\$EDITOR</code> variable as described in Setting up CVS in Your Environment.</p> <p>If you omit the <i>filename(s)</i> from your <code>commit</code> command, then CVS will recursively commit everything in the current directory. This should be avoided so that files are not inadvertently committed to the repository.</p>	<p>See the directions for Adding a File to CVS in Emacs.</p>

Updating Your Local Copy of the CVS Repository

Running CVS Update from the Command Line	Running CVS Update from Emacs
To update your local copy, go to the root of your repository and run:	Run Alt-x <code>cvs-update</code> , and enter <code>~/6.170/psN</code> when asked which directory

<p>cv^s update</p> <p>This will display a list of files that get updated. You should pay attention to the character that shows up in front of each file, as it indicates what sort of change was made to the file:</p> <ul style="list-style-type: none"> ? Indicates that this file has not been added to the repository, so CVS has not edited it. This is an indication that the file should probably either be added to the repository or to a <code>.cvsignore</code> file in that directory U Indicates that this file has been updated successfully to match the latest copy in the repository. M Indicates that your working copy of the file is different from the repository's latest version, but that merging the repository's copy with yours is successful. C Indicates that there was a conflict when trying to merge your local copy of the file with the one in the repository. See Problem Set Procedure for information about resolving CVS conflicts. P Indicates that this file has been patched which for all practical purposes is the same as updated. 	<p>to update.</p>
--	-------------------

Compiling Code and Running Programs

Compiling Code

<p>Compiling Java Code from the Command Line</p>	<p>Compiling Java Code in Emacs</p>
<p>The best way to compile code is to use Ant. We've provided a <code>build.xml</code> file that will automatically compile your code for you. All you have to do is:</p>	<p>Typing Ctrl-c Ctrl-c in Emacs should save and compile the current program. The minibuffer will prompt you for a compile command to run, which you should change</p>

1. `cd ~/6.170/psN`
2. `ant compile`

Note that if one or more of your files does not compile, you will receive an error message and no `.class` files will be generated for the `.java` file that does not compile.

to be `javac -g -d ~/6.170/psN/bin/*.java`. If you want to compile just one file, replace `*.java` with the appropriate filename.

If there are errors, typing **Ctrl-c** will put the cursor on each compilation error in turn.

If you did not allow `student-setup.pl` to change your `.emacs` file, **Ctrl-c Ctrl-c** and **Ctrl-c** will probably not work for you. In this case, you can bring up the compile dialog by running `Alt-x compile`, and you can use `Alt-x next-error` to move to the next error.

Running your program from the Command Line

Running your program from the command line is as simple as changing to the correct directory and running `java`. The following lines should do it:

1. `cd ~/6.170/psN/bin`
2. `java psN.theClassYouWantToRun`

If you get a `java.lang.NoClassDefFoundError` message, then you may not have compiled your code with the `-d ~/6.170/psN/bin` flag, as described above. If that is not the error, ask an LA for help.

Running your program from Emacs

The best way to run code from within Emacs, it turns out, is to simply open up a shell in Emacs by typing `Alt-x shell`. Then, run the two commands listed in "Running your program from the Command Line" above. If you happen to have an error in your code, middle-clicking on the error message will open the appropriate file in Emacs and jump to the right line.

Running JUnit Tests

How to run JUnit from the Command Line	How to run JUnit in Emacs
<p>To run a test suite, type the following command:</p> <pre>java junit.swingui.TestRunner <test file></pre> <p>For a non-graphical alternative method of</p>	

running JUnit, use junit.textui.TestRunner instead of junit.swingui.TestRunner.	
---	--

Editing Code in Emacs

This is a short list of some of the keyboard shortcuts in Emacs. If you haven't already, you might consider working through the Emacs tutorial: start Emacs and type `Ctrl-h t` (control-H, followed by T). You might try reading I/S's Emacs on Athena if you're unfamiliar with Emacs. Below are very common commands you are sure to use lots of times during the term.

Command (C = control, M = Alt/Meta)	Quick Name	Action
Ctrl-x Ctrl-f	New / Open	Create a new a file or open an existing file.
Ctrl-x Ctrl-s	Save	Save. Only the current active file will be saved if you have multiple files open.
Ctrl-x Ctrl-c	Quit	Quit Emacs. You will be warned if files are not saved.
Ctrl-shift-dash	Undo	Undo last command or typed text, multiple undo works
Ctrl-space	Set 'mark'	Set the 'mark' at your current cursor position. See below.
Alt-w	Copy	Copy everything from 'mark' until cursor position. Another way is to highlight the intended region with your mouse.
Ctrl-w	Cut	Cut everything from the 'mark' until cursor position.
Ctrl-y	Paste	Paste the last thing you previously cut or copied. Another way is to press the middle button on the mouse at the intended location
Ctrl-s, <i>someString</i>	Find (forward)	Move the cursor forward to the first match of <i>someString</i> . Pressing Ctrl-s again will move to the next match.
Ctrl-r, <i>someString</i>	Find (backwards)	Move the cursor backwards to the first match of <i>someString</i> . Pressing Ctrl-s again will move to the next match going backwards.

tab	Auto-Indent	the current line will be re-indented to the predicted 'correct' indentation level.
-----	-------------	--

The 6.170 staff has prepared a set of recommended add-ons to Emacs that should help you write code. These Emacs add-ons should be loaded if you ran the `student-setup.pl` script. The modifications include:

- When editing Java code, pressing `Ctrl-c Ctrl-r` will comment out the selected region, or uncomment if preceded with `Ctrl-u`, the same way `Alt-x comment-region` does normally.
- Pressing `Ctrl-h f` (mnemonic: "help for function") runs `jdk-lookup`, which displays JDK documentation for a Java package, class, method, or field.
- Middle-clicking on a Java stack backtrace takes you to the code for the clicked-upon frame.
- Use `Alt-/` to complete the partial word before point. Unlike Eclipse, this works in all buffers (including text buffers), not just in code buffers.
- Syntax highlighting (called "font locking" in Emacs) is enabled in all modes that support it, including Java mode.
- Pressing `RET` (return) not only inserts a newline but also sets up default indentation for the next line. `Ctrl-j` only inserts a newline.

Creating a New File

How to Create a New File in Emacs

Note that you perform the exact same steps to create a new file and open an existing file in Emacs. If you try to open a file that does not exist, Emacs will automatically create it for you the first time you save.

1. Use **Ctrl-x Ctrl-f** to bring up the "Find File" buffer in the bottom pane.
2. Type the fully-delimited path name of the file that you would like to create and press ENTER.