Contents

1. Overview

For Programming Languages, Part A, we will work with the Standard ML programming language (ML), using the Standard ML of New Jersey compiler (SML/NJ). You will need SML/NJ and a text editor on your computer to do the programming assignments. Any editor that can handle plain text will work, but you will want an editor that specifically supports ML with features like automatic indentation, syntax highlighting, etc. For such an editor, we recommend Emacs because we know it works and we have detailed installation instructions in this document.

You are welcome to use an editor other than Emacs. If you do, the course staff is unlikely to be able to help with any issues that arise, but you may find help on the discussion forums. While Emacs does not have the look-and-feel or tool-integration of many modern integrated development environments (IDEs), it is a versatile tool well-known by many computer scientists and software developers. If you have not used it before, you will find the menus and key bindings unusual, but learning unusual tools is commonplace in computing. Many students have used Emacs for the first time when completing this course without it being a hindrance -- we do not need any complicated features and we are writing small enough programs that you do not need to be a "power-user" with any of the tools we are using. All that said, there are plenty of people who despise Emacs for various reasons unrelated to the material in this class, so use whatever you like.

This document describes how to install, configure, and use Emacs, SML/NJ, and SML-Mode-for-emacs (henceforth SML Mode) on your computer. These instructions should work for recent versions of Windows, Mac OS X, and Linux.

These instructions were last updated in June 2016. Details and version numbers may change in the future.

Programming Languages Part B will use Racket and Programming Languages Part C will use Ruby. So there will be more software-installation in your future with instructions provided when we get there, but the instructions in this document are everything you need for Part A.

2. Emacs Installation

We strongly recommend Emacs version 24.X (for any X) so that you can use the most recent version of SML Mode. Earlier versions of SML Mode are fine, but they are more difficult to install. You can check the version of an Emacs installation in several ways, including the About Emacs option under the Help menu. Installing version 24 is easy, so we recommend doing so even if you already have an older version.

Directions depend on your operating system:

Windows:

Download a zip archive of the most recent full version, currently Version 24.5, available at http://ftp.gnu.org/gnu/emacs/windows/emacs-24.5-bin-i686-mingw32.zip. (More information and other versions of Emacs are available at the GNU Emacs website, http://www.gnu.org/software/emacs/.)

Unpack the downloaded zip archive file **`emacs-24.5-bin-i686-mingw32.zip`** by right-clicking it and choosing *Extract All*. This should produce a folder named **`emacs-24.5-bin-i686-mingw32`**. Move this folder wherever you want, but pick a permanent place (i.e., do not move it again after the next step).

Once you have moved the folder to where you want, look inside to find **`bin\addpm.exe`** (the **`.exe`** extension might not be visible in the folder window depending on how you have Windows configured). Double click on this file to run it one time and it should add a *Gnu Emacs* folder to your *Start* menu and do some other setup operations. Open the *Start* menu and select *Gnu Emacs* then *Emacs* (or just type *emacs* in the search box) to launch Emacs.

Mac OS:

Download Emacs as a Mac OS X application from http://emacsformacosx.com/. Open the disk image file (*.dmg*) and drag the Emacs application to your Applications folder. If you prefer another version of Emacs, such as the more primitive one on the command line or Aquamacs (http://aquamacs.org/), you can use it, but make sure it is based on a version of Emacs 24.X or higher.

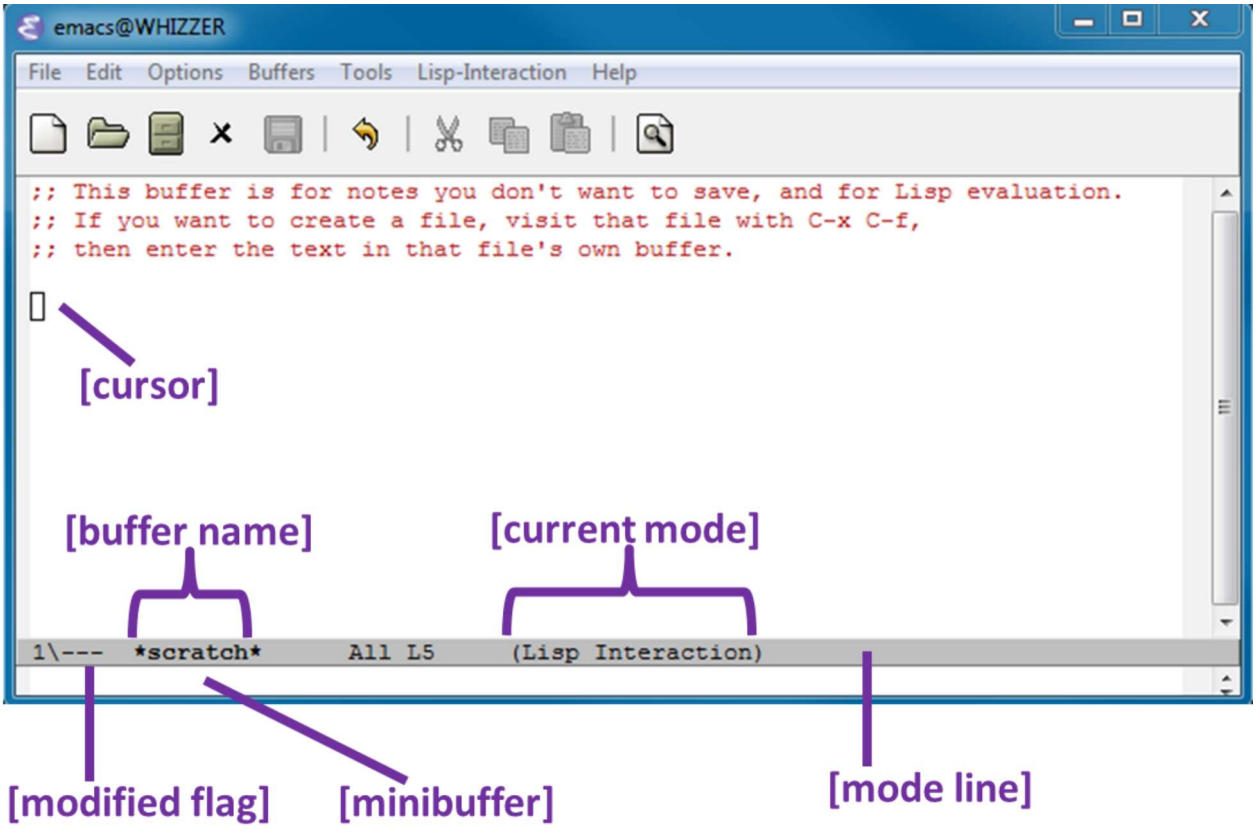
Linux:

A suitable version of Emacs is probably already installed. If not, use your package manager to install it. On Ubuntu and other Debian derivatives, try **`sudo apt-get install emacs`**. On Fedora, try **`sudo yum install emacs`**.

3. Emacs Basics

Using Emacs feels a little different than using other editors; it can take some getting used to, especially the keyboard shortcuts. Fortunately, Emacs has buttons and menus to help you adjust if that is your style. The following is a short primer on Emacs terminology and keyboard commands.

Beyond the basics described here, which should be all you need, there are countless free resources available. An introductory "tour" is at http://www.gnu.org/software/emacs/tour/. For more information, see the Emacs Reference Manual at http://www.gnu.org/software/emacs/manual/html_node/emacs/index.html (also available within Emacs from the Help Menu), the Emacs Wiki at http://www.emacswiki.org/, or the Emacs Tutorial (within Emacs from the Help Menu).



- The cursor is a rectangular block and is referred to as the **point**.

- The **mode line** displays information about the **buffer** displayed in the current **window**. A buffer is a logical "thing" that you are working on. When you open a file, you load it into a buffer, typically with the same name as the file.
- Every buffer is edited in a **mode**. The most basic mode is **Fundamental**, which provides only the most basic Emacs editing features. There are modes for many different programming languages.
- There are many "special" buffers that do not correspond to loaded files. The one above is called ***scratch***. This buffer runs in Lisp Interaction mode, which means that you can interactively type and evaluate expressions in the Lisp programming language. But we won't do that.

Emacs uses many key combinations involving the **Control** and **Meta** keys. Such key combinations are denoted **C-x** (Control-x (lowercase)) or **M-x** (Meta-x). On keyboards that don't have Meta (just about all keyboards today), ***Alt*** should work on all systems, but on some Mac systems you might use the **funny-symbol-with-four-circles** or ***Option*** keys instead. (Emacs might complain about "Super" if you get the wrong one. If you or Emacs gets confused about what you are trying to type, use **C-g** to cancel your current command and start fresh.) If none of these work, use *Esc*, but when trying to type M-x, for example, you *might* need to type Esc *then* type x. (This is only if using Esc as Meta. The other Meta "substitutes" work as usual: hold while pressing the second key.)

The most important commands in Emacs:

- **C-x C-c**: Quit Emacs
- **C-g**: Cancel the current action
- **C-x C-f**: Open a file (whether or not it already exists)
- **C-x C-s**: Save a file
- **C-x C-w**: Write a file (probably more familiar to you as *Save as...*)

Cut, copy, paste:

- Highlight text with the mouse or by hitting **C-*Space*** to *set a mark* and then moving the cursor to highlight a region.
- **C-w**: Cut a highlighted region
- **M-w**: Copy a highlighted region
- **C-k**: Cut (*k*ill) from the cursor to the end of the line
- **C-y**: Paste (*y*ank)

Some other useful commands:

- **C-x 2**: Split the window into 2 buffers, one above the other (Use the mouse or **C-x o** to switch between them)
- **C-x 0**: Undo window-splitting so there is only 1 buffer
- **C-x b**: Switch to another buffer by entering its name
- **C-x C-b**: See a list of all current buffers

Getting help within Emacs: In addition to the help button/menu on the right...

- **C-h**: Hitting this will display a short message in the minibuffer: `C-h (Type ? for further options)`.
- **C-h b**: Key *b*indings. This lists all key bindings that are valid for the current mode. Note that key bindings change from mode to mode.
- **C-h a**: Command *a*propos. After typing `C-h` a you can type a symbol and a buffer will appear that lists all symbols and functions that match that phrase.

More advanced Emacs hacks (optional):

If you are curious, try some of these once you have finished the rest of the setup instructions. They are unnecessary for any of the work we will do, but may be convenient.

- Change the colors of your syntax highlighting. In version 24 of Emacs, `M-x customize-themes` is a good place to start.
- General customization interface: Open the Options menu and choose the first item under Customize Emacs. This will let you customize Emacs through a sort-of-graphical interface. It saves all your settings in a file in your "home" directory,

`/.emacs`.

- Much, much, much more: Emacs calls itself an extensible editor for a reason.

4. SML/NJ Installation

Directions *first* vary by operating system, but then see "All Systems: Check your SML Installation" below.

Note: The latest version of SML/NJ is 110.80 and that is the version used in these installation instructions. The course videos use SML/NJ 110.75. There are no language differences between the two versions (and 110.74 and earlier are fine too), so if you already have SML/NJ 110.75 or earlier installed, that is fine. The installation instructions for Mac OS X 10.6 or later changed slightly between versions, affecting the directories where files are installed.

Windows:

Download and run the **`smlnj.msi`** installer available at http://www.smlnj.org/dist/working/110.80/. This will add an item for *SML of New Jersey* to your *Start* menu and add a command **`sml`** that you can use at the command line.

Mac OS:

The instructions here assume your operating system is Mac OS X 10.6 (which released in August 2009) or higher. See below if you have an older operating system.

Download and run the **`smlnj-x86-110.80.pkg`** installer available at http://www.smlnj.org/dist/working/110.80/. Do not use the **`.dmg`** file available; that is for older computers. We recommend you not choose a "custom install location" though you can if you adjust the instructions that follow appropriately. If you have Mac OS Sierra, you likely need 110.80 and not an older version.

Once the installation is complete, use Emacs or another text editor to edit the file **`.bash_profile`** in your home folder. (In Emacs you can do this via: **`C-x C-f /.bash_profile`**, notice the three characters "tilde, slash, dot.") If the file does not already exist, create it. Add this line to the file:

```
export PATH="$PATH:/usr/local/smlnj/bin"
```

This tells your shell (the program that you interact with in the terminal) to add the SML/NJ directory to the paths it searches to find programs. (If you are not using the bash shell, which Mac OS X has used by default since 10.3, the syntax will be different.)

Finally, you will need to run your **`.bash_profile`** to deploy the changes you have made into your environment for the present session. To do this, run:

```
source .bash_profile
```

You need to do this only once. Afterwards, each new terminal that you open will automatically run **`.bash_profile`** for you.

Now skip to "All Systems: Check your SML Installation" -- the rest of this section is for Mac users with an operating system older than OS X 10.6.

If your operating system is Mac OS X 10.5:

- If you have an Intel chip (which is the case if your computer was new in 2007 or later, or possibly in 2006), then you will need to follow the Unix instructions at http://www.smlnj.org/dist/working/110.80/.
- If you have a PowerPC chip (which is the case if your computer was new in 2005 or earlier, or possibly in 2006), then you can use the **`smlnj-ppc-110.80.dmg`** installer at http://www.smlnj.org/dist/working/110.80/. Follow the instructions above for Mac OS 10.6 or higher except (a) using the different installer and (b) making the contents of your **`.bash_profile`** file include the line:

```
export PATH="$PATH:/usr/local/smlnj-110.80/bin"
```


If your operating system is Mac OS X 10.4 or earlier: You will need to follow the Unix instructions at http://www.smlnj.org/dist/working/110.80/.

Linux:

If your package manager has a package for SML/NJ, install it. If it installs an older version such as SML/NJ 110.72, that should be fine. Otherwise, follow the Unix instructions at http://www.smlnj.org/dist/working/110.80/.

All Systems: Check your SML Installation

1. Open a terminal window and type `sml` followed by Enter/Return. To open a terminal window:

- Windows: Start then All Programs then Accessories then Command Prompt, or Windows 7/8/10 just use the Start Menu to search for the `cmd.exe` program and run it.
- Mac OS: Open Applications/Utilities/Terminal.app.
- Linux: Various ways: any shell should be fine.

2. You should see a prompt that looks like this:

```
Standard ML of New Jersey v110.80 [built: ...]
```

If you do not, then see below.

3. Make sure everything is working by typing a very simple SML program at the prompt (notice the semicolon):

```
1 + 1;
```

4. Hit *Enter/Return*. In response, the SML interpreter should print:

```
val it = 2 : int
```

5. To exit the interpreter, type *Control-Z* and then *Return* on Windows and *Control-D* on Mac or Linux.

If everything worked, skip to the next section. Else if the `sml` command caused an error, then most likely SML/NJ is installed but is not being found in your "PATH".

For **Windows**, the PATH should have been set by the installer, but if it was not for some reason, you can set it manually as follows:

- Go to Start Menu, then Control Panel, then System, then Advanced System Settings, then Advanced (the tab that should be selected by default), then Environment Variables.
- Now change the user variable `path` to be everything already there followed by:

```
;C:\Program Files (x86)\SMLNJ\bin
```

For **Mac OS X** (or Linux), double-check that you edited your `.bash_profile` file correctly. Depending on your user settings, you may need to make the same additions to a file that is in the same directory as `.bash_profile` but is called `.bashrc` or `.profile` instead. (This is particularly likely if you have MacPorts installed.)

5. SML Mode for Emacs Installation

SML Mode is an extension to Emacs that is not Emacs itself or SML/NJ itself. It displays SML code nicely with syntax coloring and clean indentation, and provides a way to run SML from within Emacs. (Thanks to Stefan Monnier for maintaining SML Mode. The website is http://www.iro.umontreal.ca/~monnier/elisp/, but you do *not* need to go there to install SML Mode.)

To install the current version of SML Mode (currently 6.7), follow these instructions from within Emacs:

- Run the command `M-x list-packages` (and then *Return/Enter*). If the `list-packages` command does not exist, your Emacs version is too old. You can upgrade Emacs or follow more difficult SML Mode installation instructions below.
- Find `sml-mode` and click on it with your mouse. If that worked, then click on `install` with your mouse then move to the next step. If you could not find `sml-mode`, first note that while package names are mostly alphabetized, they may be in more than one group, making it seem like `sml-mode` is not present. Check the entire buffer. You can most easily search using `C-s` in Emacs. If you still do not see `sml-mode`, try killing the buffer (`C-x k`) and trying the previous step again (some users have reported having to try several times, frustratingly). If you *still* do not see SML Mode, then you can follow these more manual steps instead:

1. Visit http://elpa.gnu.org/packages/sml-mode.html .
2. Locate, and download the latest version (currently sml-mode-6.7.el) from that page.
3. In Emacs type `M-x package-install-file ENTER`.
4. At the prompt `Package file name:` give the path to the just downloaded `sml-mode-6.7.el`, and type `ENTER`.
5. This will split the window, and show the `*Compile-log*` with some lines about `Compiling file`, and perhaps a warning. If there are no errors, `sml-mode` should now be installed.

- Exit and restart Emacs.
- Read below to see if you need to follow a couple more steps (more likely under Mac and Linux).

To verify that SML Mode is properly installed, let us check that it does indentation/coloring for SML files and that you can create the SML read-eval-print-loop (REPL) from within Emacs.

First, edit an existing or new SML file (try `C-x C-f test.sml` to create a new file if nothing else is handy). You should see the mode display at the bottom of the Emacs window change from Fundamental (or whatever it was) to SML. If you enter a line of code like `val n = 1;` you should see colors highlighting the keywords and variable names. When you are editing code, whenever you hit the *Tab* key, Emacs will try to reindent the current line appropriately.

Second, while the cursor is in an SML buffer (i.e., you are editing an SML file), run `C-c C-s` and press *Enter/Return*. This should split the window and create an SML prompt in a new buffer. In that buffer, you should be able to type `1+1;` and then *Enter/Return* at the prompt and see 2 as the result.

If you are seeing syntax highlighting, but the `C-c C-s` command fails with an error message, Emacs is probably having trouble finding the SML program. You can hopefully fix this as follows:

- **Mac OS**: In Emacs, edit your `.emacs` file by `C-x C-f /.emacs` (that is tilde, slash, dot, emacs) to open the file. Paste in these lines if you have Mac OS X 10.6 or later:

```
(setenv "PATH" (concat "/usr/local/smlnj/bin:" (getenv "PATH")))
(setq exec-path (cons "/usr/local/smlnj/bin"  exec-path))

```

and these lines if you have Mac OS X 10.5:

```
(setenv "PATH" (concat "/usr/local/smlnj-110.80/bin:" (getenv "PATH")))
(setq exec-path (cons "/usr/local/smlnj-110.80/bin"  exec-path))
```

(Adjust the lines above accordingly if you installed SML/NJ in a different directory.) Save the file (`C-x C-s`). Exit and restart Emacs.

- **Linux**: Find where `smlnj-110.80` was installed. Then follow the Mac OS instructions above, but replacing `/usr/local` with the appropriate path.

6. Manual SML Mode Installation for Older Emacs Versions -- Probably Skip This!

If you are using Emacs 24.X, you should be able to skip this section. These instructions are only for people who want to use an older version of Emacs, which requires using an older version of SML Mode and following different installation instructions.

If you are on Linux, your package manager may include an **`sml-mode`** package. For example, on Ubuntu, the command **`sudo apt-get install sml-mode`** may be all you need and you might be able to skip the rest of this section. Otherwise...

Go to http://www.iro.umontreal.ca/~monnier/elisp/ and download version 5.0 (see link "Download 5.0" or just download directly (http://www.iro.umontreal.ca/~monnier/elisp/sml-mode-5.0.tar.gz). Unpack it:

- Windows: You need some program that can handle a **`.tar.gz`** file. The Cygwin tools (http://www.cygwin.com) suffice, or an easier-to-use program is 7-Zip (http://www.7-zip.org/). For 7-zip, install it. Then right-click the **`.tar.gz`** file and choose *7-Zip* then *Extract Here*. If this creates a **`.tar`** file instead of a folder, do *7-Zip* then *Extract Here* on that file as well.
- Mac OS: double-click the **`.tar.gz`** file.
- Linux: Either:

1. When downloading, choose *Open with [Archive Manager]* if given the option. Otherwise, save the file and double-click it, which should open it in Archive Manager. Drag the **`sml-mode-5.0`** folder to the Desktop or wherever you want to store it.
2. Save the file. In a terminal, **`cd`** into the directory where you saved it and run **`tar -xfz sml-mode.tar.gz`**.

You should get a folder named **`sml-mode-5.0`**. You can place this folder anywhere you like.

To get SML Mode to work with Emacs, you need to create/edit a **`.emacs`** file in your "home" directory/folder that Emacs searches when it starts up. Open this file in Emacs via the command **`C-x C-f ~/.emacs`**.

Add the following line to the **`.emacs`** file:

```
(load "/path/to/where/you/stored/sml-mode-5.0/sml-mode-startup")

```

but replacing path/to/where/you/stored with the appropriate path:

- Windows: Hold down the *Shift* key as you right-click the sml-mode-5.0 folder and choose *Copy As Path*. Paste this into Emacs. Then replace each \ with / because \ is a special character in Emacs settings files. And add /sml-mode-startup at the end. So you might end up with something like Z:/look/here/sml-mode-startup.
- Mac OS: Select the folder in the Finder and choose *File* then *Get Info*. Copy the *Where:* field and add /sml-mode-startup. Now also add the following to your .emacs file in addition to the line above:

```
(setenv "PATH" (concat "/usr/local/smlnj/bin:" (getenv "PATH")))
```

```
(setq exec-path (cons "/usr/local/smlnj/bin"  exec-path))
```

(But if you have Mac OS X 10.5, use smlnj-110.77 in place of smlnj in the two lines above. Or, more generally, adjust the lines above appropriately if you installed SML/NJ in a different directory.)

- Linux: If SML Mode seems to work, but you cannot get the SML REPL to run in Emacs, then find where smlnj-110.77 was installed and add the two lines under the Mac OS instructions above, but replacing /usr/local with the appropriate path.

(There are much more elaborate installation instructions in the documentation included in the sml-mode-5.0 folder, but these are aimed at system administrators who might want to install SML Mode for an entire site instead of a single user. You do not need these.)

Now verify that SML Mode is properly installed, following the instructions in the previous section.

Finally, if you are in SML Mode but your SML code does not have syntax coloring, add this line to your ~/.emacs file and then restart Emacs:

```
(global-font-lock-mode t)
```

This should be necessary only if your Emacs version is older than 23.X.

7. Using the SML/NJ REPL (Read-Eval-Print Loop) in Emacs

At this point, we are done installing! This section shows you how to run SML programs from within Emacs. It assumes you already have an SML file or can write your own SML program in a new one.

- Edit a file with extension `.sml`. You should be in SML-mode, using *Tab* to indent your code well.
- To create the `*sml*` buffer (which holds the REPL), type `C-c C-s` (and then *Return/Enter*) in the buffer with the .sml file. (Note: This will not work in the `*scratch*` buffer that Emacs starts in because this buffer is not in SML Mode.)
- Keep the .sml file(s) you are working with for a particular assignment in the same folder. When you type `C-c C-s` to start the REPL from a buffer for foo.sml, the REPL will look in the right folder for `foo.sml` when you type `use "foo.sml";` and will look in the same folder for any other file you use such as `foo_tests.sml`. This is less confusing than trying to keep track of different folders and paths while using the REPL although that is possible.
- To end and restart a REPL session, type `C-d` (to end it) and `C-c C-s` (and then *Return/Enter*) (to restart it). You must type `C-d` while in the `*sml*` buffer; you can type `C-c C-s` from the `*sml*` buffer or a buffer with a .sml file.
- By ending and restarting a session, the new session has an empty environment. Your earlier interactions are still in the `*sml*` buffer, so you can save them, cut-paste them, etc., but they have no effect on the evaluation in the restarted REPL session.
- Evaluation can go into an infinite loop.

1. This has likely occurred if you are not getting the "-" prompt back and nothing appears to be happening.
2. `C-c C-c` will interrupt evaluation and get you your prompt back.

- If you forget to end your binding with a ";" character, the REPL will print an "=" character on the next line, which is just its way of saying, "you are not done -- continue your binding," so type a ";" and hit *Return/Enter*. This is not an infinite loop (nothing is being evaluated; the REPL is waiting for you) so `C-c C-c` does not do anything.
- If the printed result looks "pretty good," but part of what you expected to see has been replaced by a "#" or " ..." this is normal. The REPL has a limit on how many characters it prints, which is good since you might make a large value, such as a list with tens of thousands of elements. You can adjust the limit if you want.
- Two keyboard commands are particularly useful in the REPL: `M-p` will print the previous line you used in the REPL, which you can then run again or edit before running. Repeating `M-p` will cycle through previous REPL lines, allowing you to bring back up any of your previous REPL expressions. The p stands for *previous*. `M-n` (repeatedly) does the same thing in the opposite direction, with the n standing for *next*.

Advice You Will Wish You Followed!

In each REPL session, follow this pattern:

- First type `use "foo.sml";` for any SML files you want to use.
- Then use the REPL manually as long as you wish.
- After using the REPL to test something, do *not* use use to load (or reload) any more files.
- When tempted to violate the previous point, end and restart your REPL session before continuing.

Why: `use "foo.sml";` has a very simple semantics: it adds the ML bindings in the file to the environment in order. These may or may not shadow ML bindings from the last time you typed `use "foo.sml";`, depending on how foo.sml changed. This confuses even expert programmers until they train themselves to follow the pattern above.

If you find yourself typing the same non-trivial things repeatedly in the REPL, stop wasting your time.

- Move the repeated parts to a second file, e.g., `test.sml`.
- Then, when you restart your session, begin with `use "foo.sml"; use "test.sml";`.
- In fact, there is an even faster way:

1. Begin test.sml with the expression `use "foo.sml";`
2. Then begin your session with `use "test.sml";`

Note: Do *not* put `use "foo.sml";` in `test.sml` *and* begin your session with `use "foo.sml"; use "test.sml";`. That will evaluate the ML bindings in `foo.sml` twice, which is confusing.

If you develop some emotional attachment to the transcript of your `*sml*` buffer, you can save it to a file just like any other buffer. But after you do, it is not an `*sml*` buffer anymore, so you will have to create a new `*sml*` buffer from a buffer in SML Mode via `C-c C-s`.

Acknowledgments: These instructions were prepared starting with material created by Ben Wood, adapted from prior materials by Dan Grossman and Hal Perkins. Stefan Monnier provided fantastic feedback on this document and even created SML Mode version 6 to simplify SML Mode installation substantially. John Reppy provided excellent improvements to the SML/NJ installation instructions.

Mark as completed