

# Using **emacs** To Write SML

Last Revised: 4th September 2011

The easiest way to develop SML is a text editor and a way to interact with the REPL. The two obvious choices for text editor are **emacs** and **vim**. Using **vim** for SML is not different from using **vim** for anything else, because it focuses on editing text and doesn't offer any support for interaction with the REPL. **emacs** does have such support, but you have to set it up, and it takes a little getting used to.

This document will describe how to set up **emacs** for working on SML through use of **sml-mode** on a UNIX machine. By the end, you should have syntax highlighting, indentation support, and a convenient way to interact with the REPL.

## 1 Downloading **sml-mode**

**sml-mode** is distributed as a handful of elisp files, where elisp is a variant of lisp used specifically to configure **emacs**. By convention, elisp files typically live in a hidden subdirectory of your home directory at `~/.elisp`.

1. Run `mkdir ~/.elisp` to make the directory if it's not there.
2. Run `cd ~/.elisp` to change into the directory.
3. You'll now download the source for **sml-mode**. It's distributed through SVN, so running the following command will do it:

```
svn co --username anonsvn --password anonsvn  
https://smlnj-gforge.cs.uchicago.edu/svn/smlnj/sml-mode/trunk sml-mode
```

If you don't have SVN installed, you can download and extract the tarball from

<http://www.iro.umontreal.ca/~monnier/elisp/sml-mode.tar.gz>

but SVN is probably easier if you already have it.

## 2 Installing sml-mode

You'll have to edit your `emacs` configuration file to get `emacs` to use `sml-mode`. To do this, edit the file `~/.emacs` to include the following text, where `ANDREW` in the second line is replaced with your `andrew` ID:

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

(add-to-list 'load-path (expand-file-name "~ANDREW/.elisp/sml-mode/"))

(autoload 'sml-mode "sml-mode" "Major mode for editing SML." t)
(autoload 'run-sml "sml-proc" "Run an inferior SML process." t)
(setq sml-program-name "/usr/local/bin/sml")
(add-to-list 'auto-mode-alist '("\\.\\(sml\\|sig\\)\\'" . sml-mode))
(defun my-sml-mode-hook () "Local defaults for SML mode"
  (setq sml-indent-level 2)
  (setq words-include-escape t)
  (setq indent-tabs-mode nil))
(add-hook 'sml-mode-hook 'my-sml-mode-hook)
```

If you already have things in this file, you can append the above to the bottom of the file.

## 3 Using sml-mode

With just the above and no further action on your part, `emacs` will load SML code with syntax highlighting and hitting the `tab` key will (usually) well indent your code.

`sml-mode` is also a great way to interact with the REPL, though.

- `C-x 3` splits your current window in half vertically.
- `C-x o` switches your cursor from one window-half to the other.
- `C-c C-c` tells `sml-mode` that you'd like to compile whatever file is in the window-half with your cursor. `emacs` will then prompt you for a compile command. `CM.make("sources.cm")` is usually what you want, unless the CM file is called something else. It will then ask you what the command that loads SML is.

Once you confirm the location of your `sml` binaries, `sml-mode` will run a fully-fledged REPL in the other window-half. You can type SML expressions at it like you would a REPL from the command line.

If your code has errors in it, the line numbers will appear in red in the REPL window. If you put your cursor on that that red section of text and hit `enter`, it will jump your cursor to the problematic line in the other window.

## 4 More Information

There's a truly terrifying amount of information about how to use and customize `emacs` on the internet. Here are some links, but there are many more.

- <http://www.smlnj.org/doc/Emacs/sml-mode.html>
- <http://www.iro.umontreal.ca/~monnier/elisp/>
- <http://www.emacswiki.org/>
- <http://www.emacswiki.org/emacs/SMLMode>