

Help Topic: The SVN Move Command

Spring 2020

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

Useful emacs commands

Emacs contains a huge wealth of built in functionality to make basic tasks easier. If you ever find yourself asking "can emacs do X?", the answer is probably "Yes! And it has a keyboard shortcut!" It is possible to do nearly every editing task using your keyboard.

Emacs Help

Emacs has a powerful help system. The help system will easily show you information about how to use a command, what key it is bound to (or what key it would be bound to if you've stolen the binding for something else). Help can be invoked with `C-h` followed by another key depending on what type of help you want.

Important help commands: `C-h ?` - Help on help.

`C-h k` - Help on keys. Type this command, followed by a keyboard shortcut. Help for the function bound to that key sequence is displayed. For example, `C-h k C-h k` will tell you that

```
C-h k runs the command describe-key
```

`C-h f` - Help on functions. Type a function name to find out about it. If you press `tab`, emacs will offer completion suggestions, so you don't necessarily need to know the name of the function.

`C-h v` - Help on variables. Like `C-h f`.

`C-h m` - Help for mode. Find out features of the current major mode and any active minor modes. Helpful for discovering useful functions that the mode's authors have included and bound to keys.

`C-h i` - Emacs info. Browseable help manual.

C++ Programming in Emacs

Opinions differ on the use of tabs in source code, but if you want to avoid inserting them, you can put the following in `.emacs`. You can change "2" in the second line to be whatever tab width you prefer:

```
(setq-default indent-tabs-mode nil)
(setq tab-width 2)
```

By default, emacs open `.h` files in C-mode instead of C++ mode. You can improve your syntax highlighting by switching to C++ mode (`M-x C++-mode RET`), but if you want to make it default, use the following:

```
(add-to-list 'auto-mode-alist '(("\\.h\\\\" . c++-mode))
```

As long as a class implementation (.cpp file) and header (.h file) are in the same folder, emacs will easily switch between them for you with the command: `M-x ff-find-other-file RET`. This is most useful if you assign it to a shortcut key. You can put something like the following in your .emacs:

```
(global-set-key (kbd "C-f") 'ff-find-other-file)
```

You can make certain words get special syntax highlighting, even in comment strings:

```
(add-hook 'c-mode-common-hook
  (lambda ()
    (font-lock-add-keywords nil
      '(("\\<\\(FIXME\\|TODO\\|HACK\\|fixme\\|todo\\|hack\\)" 1
        font-lock-warning-face t))))))
```

Directory locals are a powerful tool to customize the way emacs treats an entire directory. For example, given the following directory structure:

```
project/
  build.sh
  src/
    pApp/
    lib_foo/
```

Creating the file `src/.dir-locals` will direct emacs to apply the settings in that file to all files in the `src` directory.

For example, if `.dir-locals` contains

```
((c++-mode
  . ((compile-command . "cd ../../\&; ./build.sh"))))
```

Then for any file in `c++-mode`, using the emacs `compile` command will invoke the build script. You might want to bind compilation to a key and make the compilation buffer scroll to show any perti-

nent errors:

```
(global-set-key (kbd "<f5>") 'compile)
(setq compilation-scroll-output t)
```