

Bogdan's EMACS Config

Bogdan Popa

May 13, 2014

Contents

1	This Document	3
2	Installation	3
2.1	Linux	3
2.2	Mac OS X	3
2.3	Windows	4
3	Setup	4
3.1	Unix	4
3.2	Windows	4
4	Bindings	4
4.1	General	4
4.2	Getting Help	5
4.3	Moving Around	5
4.4	Buffer Management	5
4.5	Window Management	6
4.6	Popups	6
5	Bookmarks	6
5.1	Bindings	6
5.2	EVIL Bindings	6
6	Paredit	7
7	Imenu	7
8	Backups	7

9	dired+	7
9.1	Bindings	7
10	EVIL	8
10.1	Differences From Standard EVIL	8
10.2	Common Bindings	8
10.3	Normal Mode Bindings	8
10.4	Visual Mode Bindings	8
10.5	Insert Mode Bindings	8
11	Magit	9
11.1	Generic Bindings	9
11.2	Status Bindings	9
11.3	Branch Bindings	9
12	Org	9
12.1	EMACS Bindings	9
12.1.1	Agenda	9
12.1.2	Buffer	10
12.1.3	Document	10
12.1.4	Headlines	10
12.1.5	Movement	10
12.1.6	Source Code	10
12.2	EVIL Bindings	11
12.2.1	Agenda	11
12.2.2	Headlines	11
12.2.3	Movement	11
13	Prodigy	11
13.1	Services	11
13.2	Bindings	12
14	Haskell	12
14.1	Structured Haskell Mode	12
14.2	Bindings	12
15	PHP	12
15.1	Flycheck	13

16 Python	13
16.1 Flycheck	13
16.1.1 Ignoring Certain Errors	13
16.1.2 Bindings	13
16.2 Jedi	13
16.2.1 Bindings	13
17 Scala	13
18 Scheme	14
18.1 Setup	14
18.2 Geiser mode	14
18.2.1 Buffer Bindings	14
18.2.2 REPL Bindings	14
19 Clojure	14
19.1 Setup	14
19.2 Cider	14
19.2.1 Buffer Bindings	15
19.2.2 REPL Bindings	15

1 This Document

The purpose of this document is mostly to help me remember all of my settings as well as to help newbies set up their environment and get started using my configuration.

2 Installation

2.1 Linux

This depends on the distribution. Most of them will have EMACS in their repositories.

```
sudo apt-get install emacs-snapshot-gtk
```

2.2 Mac OS X

Download and install EMACS for OS X from EMACS for Mac OS X.

2.3 Windows

Download and install EMACS for Windows from the GNU Repositories.

3 Setup

3.1 Unix

Clone the repo into your home directory and then fetch its submodules.

```
cd ~/
git clone https://github.com/Bogdanp/.emacs.d.git
cd .emacs.d
git submodule init
git submodule update
```

3.2 Windows

Windows makes things a bit harder. Add an environment variable called `HOME` and make it point to `C:/Users/Username/`. Clone the `windows` branch of the repository in that folder and fetch its submodules.

4 Bindings

The most important EMACS binding is `C-g`. Use it as an escape hatch for when you mess up your key combinations. VIM users: `C-g` is your `ESC` in EMACS.

4.1 General

`M-x` Bring up extended command mode.

`C-x C-c` Quit.

`C-x C-i` Bring up `imenu`.

`C-x C-e` Evaluate last sexp.

4.2 Getting Help

C-h f Describe function.

C-h v Describe variable.

C-h k Describe key combination.

4.3 Moving Around

C-s Search forward.

C-p Goto prev line.

C-n Goto next line.

C-a Goto beginning of line.

C-e Goto end of line.

C-b Goto prev character.

C-f Goto next character.

M-b Goto prev word.

M-f Goto next word.

M-< Goto beginning of buffer.

M-> Goto end of buffer.

C-w Kill word backward.

M-d Kill word.

4.4 Buffer Management

C-x C-s Save.

C-x C-w Save as.

C-x C-b List buffers.

C-x b Switch buffers.

C-x f Open.

4.5 Window Management

These suck compared to the EVIL alternatives but they (mostly `C-x o`) will come in handy when dealing with `<E>` buffers.

`C-x 0` Delete current window.

`C-x 1` Delete other windows.

`C-x 2` Split window vertically.

`C-x 3` Split window horizontally.

`C-x o` Goto other window.

`C-c M-a` toggle between the current window state and a fullscreen terminal.

4.6 Popups

`q` Will close most popups, make sure you are in EMACS mode before you use it (you're in EMACS mode if there's an `<E>` in your status bar).

5 Bookmarks

You can set bookmarks to any line in any buffer (incl. dired buffers).

5.1 Bindings

`C-x r m` Create bookmark.

`C-x r b` Goto bookmark.

`C-x r l` List bookmarks.

5.2 EVIL Bindings

`, bc` Create bookmark.

`, bb` Goto bookmark.

`, bl` List bookmarks.

6 Paredit

If you're writing any sort of LISP you **need** to learn Paredit. It's that good. Paredit allows you to perform structured editing of sexps. Read more about it on the EMACS Wiki.

NOTE: If you're coming from VIM Paredit will probably leave you frustrated (there will be times when you delete a closing paren in normal mode and then paredit won't allow you to insert any closing parens) if you don't take the time to learn it. So make sure you do.

You can find a Paredit cheatsheet [here](#).

7 Imenu

Imenu produces an index of all of the definitions in the current buffer and allows to you easily go to any of them. It is bound to `C-x C-i`.

8 Backups

`undo-tree-mode` is enabled globally. All backups - including `undo-tree` files - are saved in `temporary-file-directory`.

9 dired+

You can bring up dired by pressing `C-j` on a folder whilst in `ido-find-file` (`C-x C-f`). You can also bring it up with `C-x d`.

9.1 Bindings

- `n` Goto next line.
- `p` Goto prev line.
- `+` Create directory.
- `x` Execute changes.
- `D` Delete file immediately.
- `d` Mark file for deletion.
- `g` Refresh buffer.

m Mark file.
u Unmark file.
R Rename file(s).
U Unmark all files.

10 EVIL

10.1 Differences From Standard EVIL

My EVIL mode fork makes some changes to the way registers are handled. Most importantly, while you are inside an EVIL mode buffer you can only access the clipboard through the + register and nothing else, this prevents EMACS from dirtying the clipboard whenever you cut things.

The copy-on-motion bug is fixed.

C-w now works in the minibuffer.

10.2 Common Bindings

You can drop into EMACS mode for a single command using \.

C-z Toggle between EVIL and EMACS mode.

10.3 Normal Mode Bindings

SPC Bring up ace jump mode.

S-SPC Bring up ace char jump mode.

C-w f Toggle window fullscreen.

10.4 Visual Mode Bindings

The **C-a**, **C-e**, **C-p**, **C-n** bindings are available in visual mode.

10.5 Insert Mode Bindings

The **C-a**, **C-e**, **C-p**, **C-n** bindings are available in insert mode.

C-w Kill word backward.

C-r Yank from register.

11 Magit

Magit is fucking great. You can find its official manual [here](#).

11.1 Generic Bindings

n Goto next object.

p Goto prev object.

TAB Expand/collapse object.

RET Open object.

11.2 Status Bindings

Use **C-c m** to bring `magit-status` up.

S Stage everything.

s Stage object under point.

u Unstage object under point.

b* Branch operations.

c* Commit operations.

f* Fetch operations.

F* Pull operations.

P* Push operations.

11.3 Branch Bindings

Use **bv** inside `magit-status` to bring up the visual branch manager.

12 Org

12.1 EMACS Bindings

12.1.1 Agenda

C-c . Insert date.

C-u C-c . Insert date and time.

12.1.2 Buffer

C-x n s Narrow buffer.

C-x n w Widen buffer.

12.1.3 Document

C-c C-e Export document.

12.1.4 Headlines

TAB Expand/collapse headline.

S-TAB Expand/collapse everything.

C-RET Add new headline at the current level.

M-RET Add new headline/list item at the current level.

M-→ Increase headline level.

M-← Decrease headline level.

S-M-→ Increase headline level incl. children.

S-M-← Decrease headline level incl. children.

12.1.5 Movement

C-c C-u Goto parent headline.

C-c C-b Prev same-level headline.

C-c C-f Next same-level headline.

C-c C-p Prev headline.

C-c C-n Next headline.

12.1.6 Source Code

C-c' Open up the code in a source code block in a separate buffer for editing.

<sTAB Insert source code block. You need to be in insert or EMACS mode to use this one.

12.2 EVIL Bindings

12.2.1 Agenda

, a Bring up agenda (this one is globally-available).

12.2.2 Headlines

, t Toggle headline state.

- Cycle list bullet.

< Decrease headline level.

> Increase headline level.

12.2.3 Movement

gu Goto parent headline.

gj Goto next same-level headline.

gk Goto prev same-level headline.

13 Prodigy

Prodigy is bound to C-c p.

13.1 Services

Services should be defined in `config/init-prodigy.el`. The following is an example service:

```
1: (prodigy-define-service
2:   :name "Python Server"
3:   :command "python"
4:   :args '("-m" "SimpleHTTPServer" "8093")
5:   :cwd "/path/to/my/project/"
6:   :tags '(example)
7:   :kill-process-buffer-on-stop t)
```

13.2 Bindings

- n** Next process.
- p** Prev process.
- s** Start process.
- S** Stop process.
- r** Restart process.
- \$** View process log.
- F** Clear filters.
- f t** Filter by tag.
- f n** Filter by name.
- j d** Open dired for the current process.
- j m** Open magit for the current process.

14 Haskell

You will need a recent version of `ghc` and `cabal` as well as the following packages: `hlint`, `ghc-mod`, `ghci-ng`, `structured-haskell-mode`, `stylish-haskell`.

14.1 Structured Haskell Mode

See the Official Repository.

14.2 Bindings

- C-c C-l** Load the buffer into a REPL.
- C-c C-t** View type of object at point.

15 PHP

PHP is supported through `php-mode` and `web-mode`. Use `web-mode` for mixed HTML and PHP files and `php-mode` for pure PHP files. The default mode for PHP files is `web-mode`.

15.1 Flycheck

Install `phpcs` from PEAR and you should be good to go.

16 Python

16.1 Flycheck

Install `flake8` to use it as a backend for Flycheck.

16.1.1 Ignoring Certain Errors

Create a `.flake8rc` file in your `HOME` directory. For example:

```
1: [flake8]
2: ignore = E501,F403,E712
```

16.1.2 Bindings

`C-c !n` Goto next error.

`C-c !p` Goto prev error.

`C-c !l` List errors.

16.2 Jedi

Install `virtualenv` and `epc` and then run `M-x jedi:install-server`.

16.2.1 Bindings

`C-c .` Goto definition (and `C-c ,`).

`C-c ?` Show documentation of the object at point.

17 Scala

View the ENSIME manual at the Official Repository. Start up ENSIME in a Scala buffer with `M-x ensime RET`.

18 Scheme

18.1 Setup

Link `mzscheme` and `racket` so that they can be found in `PATH`.

18.2 Geiser mode

Start it up with `M-x run-geiser`.

18.2.1 Buffer Bindings

`C-c C-a` Switch to the REPL and enter the current module.

`C-c C-z` Switch between the buffer and the REPL.

`C-M-x` Eval definition around point.

`C-c M-e` Eval definition around point and switch to REPL.

`C-c C-x` Eval sexp before point.

`C-c C-r` Eval region (also `C-c M-r`).

`C-c C-b` Eval buffer (also `C-c M-b`).

`M-g n` Jump to next error.

`M-g p` Jump to prev error.

18.2.2 REPL Bindings

`C-c C-q` Kill Scheme process.

`C-c M-o` Clear REPL.

19 Clojure

19.1 Setup

Install `leiningen`.

19.2 Cider

Start it up with `M-x cider-jack-in`.

19.2.1 Buffer Bindings

C-c C-z Switch between the buffer and the REPL.

C-c C-x Eval sexp before point.

C-c C-r Eval region (also **C-c M-r**).

C-c C-b Eval buffer (also **C-c M-b**).

C-c M-e Eval definition around point and switch to REPL.

C-M-x Eval definition around point.

19.2.2 REPL Bindings

C-c M-o Clear REPL.