

Bogdan's EMACS Config

Bogdan Popa

April 27, 2014

Contents

1	This Document	2
2	Installation	2
2.1	Linux	2
2.2	Mac OS X	2
2.3	Windows	3
3	Setup	3
3.1	Unix	3
3.2	Windows	3
4	Bindings	3
5	Backups	4
6	EVIL	4
6.1	Differences From Standard EVIL	4
6.2	Common Bindings	4
6.3	Normal Mode Bindings	4
6.4	Visual Mode Bindings	4
6.5	Insert Mode Bindings	4
7	Prodigy	5
7.1	Services	5
7.2	Bindings	5

8	Magit	5
8.1	Generic Bindings	6
8.2	Status Bindings	6
8.3	Branch Bindings	6
9	Python	6
9.1	Flycheck	6
9.1.1	Ignoring Certain Errors	6
9.1.2	Bindings	7
9.2	Jedi	7
9.2.1	Bindings	7
10	Scheme	7
10.1	Setup	7
10.2	Geiser mode	7
10.2.1	Buffer Bindings	7
10.2.2	REPL Bindings	8

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.

- `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).
- `C-p` and `C-n` goto next and prev line.
- `C-a` and `C-e` goto the beginning and end of the current line.
- `M-x` command mode.
- `C-x b` switch buffer.
- `C-x f` find file.
- `C-x o` switch to other window.
- `C-c M-a` toggle between the current window state and a fullscreen terminal.

5 Backups

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

6 EVIL

6.1 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` works in the minibuffer and in the search buffer.

6.2 Common Bindings

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

- `C-z` switch between EVIL and EMACS mode.

6.3 Normal Mode Bindings

- `SPC` bring up ace jump mode.
- `S-SPC` bring up ace char jump mode.
- `C-w f` toggle between making the current window fullscreen or not.

6.4 Visual Mode Bindings

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

6.5 Insert Mode Bindings

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

- `C-w` deletes the previous word.
- `C-r` inserts whatever is in a given buffer.

7 Prodigy

Prodigy is bound to C-c p.

7.1 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)
```

7.2 Bindings

- `s` start a process.
- `S` stop a process.
- `r` restart a process.
- `$` open a 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.

8 Magit

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

8.1 Generic Bindings

- `n` goto next object.
- `p` goto previous object.
- TAB expand/collapse object.
- RET open object.

8.2 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.

8.3 Branch Bindings

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

9 Python

9.1 Flycheck

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

9.1.1 Ignoring Certain Errors

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

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

9.1.2 Bindings

- C-c !n and C-c !p goto next and prev error.
- C-c !l list errors.

9.2 Jedi

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

9.2.1 Bindings

- C-c . goto definition (and C-c ,).
- C-c ? show documentation of the object at point.

10 Scheme

10.1 Setup

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

10.2 Geiser mode

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

10.2.1 Buffer Bindings

- C-c C-a to switch to the REPL and enter the current module.
- C-c C-z to 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 and M-g p to jump to next and prev error.

10.2.2 REPL Bindings

- C-c C-q kill Scheme process.
- C-c M-o clear REPL.