

# Contents

<b>1</b>	<b>Installation</b>	<b>2</b>
<b>2</b>	<b>Package Setup</b>	<b>3</b>
2.1	Gnus . . . . .	3
2.2	Term . . . . .	3
<b>3</b>	<b>Functionality</b>	<b>4</b>
3.1	Code Tagging and Navigation . . . . .	4
3.2	Code Completion . . . . .	4
<b>4</b>	<b>Structure</b>	<b>5</b>
<b>5</b>	<b>Best Practices</b>	<b>6</b>
<b>6</b>	<b>To-Do</b>	<b>7</b>
6.1	Customize Mode Line . . . . .	7
6.2	Disable cursor in pdf-view-mode . . . . .	7
6.3	Get much better at Gnus . . . . .	8
6.4	Company Backends . . . . .	8
6.5	load-file from Emacs . . . . .	8
6.6	align-current AUCTeX . . . . .	8
6.7	align uses 2 spaces in Emacs . . . . .	8
6.8	When Emacs opens it is not completely maximized . . . . .	8
6.9	Single quote delimiter not working . . . . .	8
6.10	Command that closes all buffers for which underlying file was deleted . . . . .	8
6.11	run-octave opens in other buffer . . . . .	8
6.12	Magit bug missing headers . . . . .	9
6.13	Verilog mode escape . . . . .	9
6.14	Verilog code tagging . . . . .	9
6.15	Better Code Tagging . . . . .	9

# Chapter 1

## Installation

I've fetched Emacs from the [GitHub mirror repository](#). The source code resides in ~/developer/software/emacs. Installation instructions reside in the source directory in the file named "INSTALL". I've chosen to use a build directory named "build". From the build directory run configure with:

```
1 $ ../../configure --with-x-toolkit=gtk3 --with-mailutils \  
2 --with-imagemagick --with-xwidgets
```

I'm currently using version 26.1, by running:

```
1 $ git checkout emacs-26.1
```

There is some issue with highlighting in version 27.050 (potentially font-lock?) that makes it very annoying to write code. In general I should only use stable releases, which are specified on the [Emacs website](#).

## Chapter 2

# Package Setup

### 2.1 Gnus

Gnus was a huge pain to setup. I finally got it working with the information at this [link](#).

### 2.2 Term

Ansi term uses 8 colors for the terminal GUI. We can redefine these colors in an Emacs terminal by customizing their values with `M-x customize-group RET term RET`. The colors must also be configured in `~/.bashrc`.

## Chapter 3

# Functionality

### 3.1 Code Tagging and Navigation

### 3.2 Code Completion

I'm currently using company with various backends for completion in different major modes. For C/C++ I'm using company-rtags. This potentially has the benefit of being aware of the current project (e.g. completions for functions I've defined and custom headers) although this is unconfirmed. However, it's a bit slow and some of the completions don't seem entirely accurate. Another option could be company-ycmd. I believe this is faster although it may come at the expense of being unaware of the environment. This should be verified.

## Chapter 4

# Structure

## Chapter 5

# Best Practices

[This Emacs manual page](#) contains principles to follow when customizing key bindings. Basically, `C-c <letter>` (but not `C-c` followed by another control character) as well as `<f5>` through `<f9>` are free for users to define how they wish.

# Chapter 6

## To-Do

### 6.1 Customize Mode Line

I can customize mode-line-format on a per-mode basis (see [this StackExchange question](#)). Also, see the [Emacs documentation](#) on the subject.

### 6.2 Disable cursor in pdf-view-mode

I'd like to disable the cursor when viewing a PDF. It's distracting and provides no value. I've tried these two additions to `init.el` to no avail:

```
1 (add-hook 'pdf-view-mode-hook
2           (lambda ()
3             (make-variable-buffer-local 'cursor-type)
4             (setq cursor-type nil)))
5 (add-hook 'post-command-hook
6           (lambda ()
7             (setq cursor-type (if pdf-view-mode t 'nil))))
```

These are the links I've found that address related issues:

[How to change the cursor type and color?](#)

[Cursor parameters](#)

[How can I make the cursor change to block in overwrite mode?](#)

[Changing Cursor Dynamically](#)

[Cursor Display](#)

## 6.3 Get much better at Gnus

## 6.4 Company Backends

There are various company backends that don't work. At the very least company-reftex and company-auctex aren't working. Based on info from the company-backends variable, it sounds like only one backend can be used at a time. This might be the issue.

## 6.5 load-file from Emacs

This causes cursor color to change and potentially other behavior as well.

## 6.6 align-current AUCTeX

align-current works well for tables with limited text. However, it propagates line breaks and so takes up a needless number of lines for long text sections that have to be manually adjusted. Also, I'd like to be able to run this globally for a whole buffer. The most similar action is align-entire but this aligns all tables to one another; I'd like aligning to be done on a per-table basis.

## 6.7 align uses 2 spaces in Emacs

The align environment seems to use 2 spaces in Auctex instead of the 8 used by everything else. It should be 8.

## 6.8 When Emacs opens it is not completely maximized

The right side of the frame is not quite right.

## 6.9 Single quote delimiter not working

Pressing ´ causes three single quotes to be inserted.

## 6.10 Command that closes all buffers for which underlying file was deleted

Probably better, just do this automatically. One worry is if file was accidentally deleted and buffer allows you to keep the file alive. Still, existing buffers shouldn't be used as a sort of backup.

## 6.11 run-octave opens in other buffer

`M-x run-octave` opens the octave shell in the other buffer.



## 6.12 Magit bug missing headers

I'm getting the error: "BUG: missing headers nil" when using magit with ghub. This is a known issue and is tracked [here](#). I guess the solution is to just wait for someone to fix it there.

## 6.13 Verilog mode escape

When in evil insert mode and the cursor position is after a reg or wire, pressing escape tries to vectorize the reg/wire instead of entering evil normal state.

## 6.14 Verilog code tagging

Verilog mode should have code tagging and navigation. See [this link](#) for ideas on code tagging.

## 6.15 Better Code Tagging

The problem with RTags is that it only works for C-based languages. gtags should work for many others. See [this link](#). It probably makes sense to use the [ggtags](#) interface. Also use the [helm interface](#). This may also be the wrong question, but maybe it's also possible to use rtags as a backend for a common gtags frontend?