

# My literate GNU Emacs config

Marius Peter

*23 Jul, 2020 (Thu)*

## Contents

<b>1</b>	<b>TODO First-time setup</b>	<b>4</b>
1.1	File system paths . . . . .	4
<b>2</b>	<b>Early setup</b>	<b>5</b>
2.1	Garbage collection . . . . .	5
2.2	Emacs client . . . . .	5
2.3	Custom file . . . . .	5
2.4	Profiling — start . . . . .	5
2.5	Customization shortcuts . . . . .	6
2.6	Backups . . . . .	6
2.7	Secrets . . . . .	6
<b>3</b>	<b>Global key bindings</b>	<b>6</b>
3.1	Navigation . . . . .	7
3.1.1	Graphical file open . . . . .	7
3.1.2	Exit behaviours . . . . .	7
3.2	Mouse zoom . . . . .	7
<b>4</b>	<b>Packages</b>	<b>7</b>
4.1	Package archives . . . . .	8
4.2	<code>use-package</code> . . . . .	8
4.3	<b>TODO</b> Convenient package update . . . . .	8
4.4	<code>ivy</code> . . . . .	8
4.4.1	<code>counsel</code> . . . . .	8
4.4.2	<code>swiper</code> . . . . .	9
4.5	<code>evil-mode</code> . . . . .	9
4.6	<code>org-mode</code> . . . . .	9
4.6.1	Basic customization . . . . .	9

4.6.2	Invisible edits . . . . .	10
4.6.3	Agenda . . . . .	10
4.6.4	Timestamps . . . . .	10
4.6.5	L <sup>A</sup> T <sub>E</sub> X export . . . . .	10
4.6.6	Publish . . . . .	11
4.6.7	Export . . . . .	12
4.7	<b>undo-tree</b> . . . . .	12
4.8	<b>dumb-jump</b> . . . . .	12
4.9	<b>gnuplot</b> . . . . .	12
4.10	<b>Ledger</b> . . . . .	13
4.11	<b>TODO</b> Sidebar . . . . .	13
4.12	<b>which-key</b> . . . . .	13
4.13	<b>company</b> . . . . .	13
4.14	<b>flycheck</b> . . . . .	13
4.15	<b>csv-mode</b> . . . . .	13
4.16	<b>json-mode</b> . . . . .	13
4.17	<b>magit</b> . . . . .	14
4.18	<b>pdf-tools</b> . . . . .	14
4.19	<b>dashboard</b> . . . . .	14
4.20	<b>rainbow-mode</b> . . . . .	14
4.21	<b>projectile</b> . . . . .	14
4.22	<b>all-the-icons</b> . . . . .	15
4.23	<b>Better parentheses</b> . . . . .	15
<b>5</b>	<b>Cosmetics</b> . . . . .	<b>15</b>
5.1	Cursors . . . . .	15
5.1.1	Default cursor . . . . .	15
5.2	Faces . . . . .	15
5.2.1	<b>variable-pitch-mode</b> . . . . .	16
5.2.2	<b>TODO</b> Default font size . . . . .	16
5.3	Initial and default frames . . . . .	16
5.3.1	GNU/Linux . . . . .	16
5.4	Theme . . . . .	17
5.4.1	<b>blendoit-light</b> . . . . .	17
5.4.2	<b>blendoit-dark</b> . . . . .	17
<b>6</b>	<b>Editing preferences</b> . . . . .	<b>17</b>
6.1	Clean up menus . . . . .	18
6.2	Coding standards . . . . .	18
6.3	Dividers . . . . .	18

6.4	Tabs . . . . .	18
6.5	auto-fill . . . . .	18
6.6	Recent files . . . . .	19
6.7	pop-up-frames . . . . .	19
<b>7</b>	<b>Late setup</b>	<b>19</b>
7.1	Profiling — stop . . . . .	19
7.2	Profiling — report . . . . .	19
<b>8</b>	<b>Conclusion</b>	<b>19</b>

## Abstract

GNU Emacs is most often used as a text editor. The utmost level of customisation is afforded by enabling the user to rewrite *any* part of the source code and observe the editor's modified behaviour in real time. Since its inception in 1984, GNU Emacs has grown to be much more than a full-featured, high-productivity text editor—new *modes* have been written to interact with hundreds of file formats, including `.txt`, `.pdf`, `.jpg`, `.csv`, and `.zip` just to name a few. This configuration file itself was written in *Org mode*, a collection of functions enabling the harmonious mixing of code and comments in view of publication: this is the endgame of *literate programming*.

## 1 TODO First-time setup

Spacemacs-like dialog for default settings.

Bootstrap main configuration from USB drive.

```
;; Prompt enterprise or personal install. Create file in .emacs.d/ on Linux,
;; AppData/ on Windows. Ask user for details and preferred bindings.

; Check if .emacs.d exists

; If it does, warn user

; Copy init-bootstrap.el from USB to where operating systems expects init.el

;; (defun blendoit/first-time-setup-windows-nt ()
;;   "Execute the first-time setup on MS Windows.
;; If no '.emacs.d/' config exists on local system, copy
;; init-bootstrap.el to '~.emacs.d/'."
;;   (interactive)
;;   (find-file "~/.emacs.d/blendoit/blendoit-init.org"))

;; (cond ((string-equal system-type "windows-nt") blendoit/first-time-setup-windows-nt)
;;       ((string-equal system-type "gnu/linux") blendoit/first-time-setup-linux))
```

### 1.1 File system paths

In this subsection, we tell Emacs about relevant paths to resources.

On my MS Windows machine, I add the path to Portable Git.<sup>1</sup>

```
(if (string-equal system-type "windows-nt")  
    (add-to-list 'exec-path "C:/Users/marius.peter/PortableGit/bin/"))
```

## 2 Early setup

### 2.1 Garbage collection

First, we increase the RAM threshold beyond which the garbage collector is activated.

```
(setq gc-cons-threshold 100000000)
```

### 2.2 Emacs client

Makes opening emacs faster for following instances.

```
; (setq initial-buffer-choice (lambda () (get-buffer "*dashboard*")))
```

### 2.3 Custom file

Load settings created automatically by GNU Emacs Custom. (For example, any clickable option/toggle is saved here.) Useful for fooling around with M-x customize-group <package>.

user-emacs-directory

```
(setq custom-file (concat user-emacs-directory "init-custom.el"))  
(load custom-file)
```

### 2.4 Profiling — start

We start the profiler now , and will interrupt it in section 7.1. We will then present profiling report in section 7.2.

```
; (profiler-start)
```

---

<sup>1</sup>Download from <https://git-scm.com/download/win>

## 2.5 Customization shortcuts

We begin by defining a user shortcut to this very file:

```
(defun find-init-blendoit ()
  "Jump to this very file."
  (interactive)
  (find-file my/literate-config))

(global-set-key (kbd "C-c c") 'find-init-blendoit)
```

Now, different shortcuts for other customization actions:

```
(global-set-key (kbd "C-c v") 'customize-variable)
(global-set-key (kbd "C-c f") 'customize-face)
```

## 2.6 Backups

Backups are so important that they should be described right after the shortcut to this file.

```
(setq backup-directory-alist '(("." . ,temporary-file-directory))
auto-save-file-name-transforms '(("." ,temporary-file-directory t))
  backup-by-copying t      ; Don't delink hardlinks
  version-control t        ; Use version numbers on backups
  delete-old-versions t    ; Automatically delete excess backups
  kept-new-versions 20     ; how many of the newest versions to keep
  kept-old-versions 5      ; and how many of the old
)
```

## 2.7 Secrets

```
(setq user-full-name "Marius Peter"
      user-mail-address "blendoit@gmail.com")

(setq user-full-name "Marius Peter"
      user-mail-address "blendoit@gmail.com")
```

## 3 Global key bindings

The following bindings strive to further enhance CUA<sup>2</sup> mode.

---

<sup>2</sup>Common User Access.

### 3.1 Navigation

```
(global-set-key (kbd "C-'"') 'delete-other-windows)
(global-set-key (kbd "C-s") 'save-buffer)
(global-set-key (kbd "C-b") 'ibuffer-sidebar-toggle-sidebar)
(global-set-key (kbd "C-r") 'counsel-recentf)
; (global-set-key (kbd "C-n") 'make-frame) ; 7aram!
```

#### 3.1.1 Graphical file open

Open file with C-o.

```
(global-set-key (kbd "C-o") 'menu-find-file-existing)
```

#### 3.1.2 Exit behaviours

The following bindings lead to more natural exit behaviors.

```
(defun delete-window-or-previous-buffer ()
  "Delete window; if sole window, previous buffer."
  (interactive)
  (if (> (length (window-list)) 1)
      (delete-window)
      (previous-buffer)))

(global-set-key (kbd "C-w") 'delete-window-or-previous-buffer)
(global-set-key (kbd "C-q") 'save-buffers-kill-terminal)
```

### 3.2 Mouse zoom

The typical binding on both GNU/Linux and MS Windows is adequate here:  
C-= to zoom in, C-- to zoom out.

```
(global-set-key (kbd "C--") 'text-scale-decrease)
(global-set-key (kbd "C-=") 'text-scale-increase)
(global-set-key (kbd "C-+") 'text-scale-increase)
```

## 4 Packages

Packages are collections of .el files providing added functionality to Emacs.

## 4.1 Package archives

List of package archives.

```
(require 'package)
(add-to-list 'package-archives '("melpa" . "https://melpa.org/packages/") t)
(add-to-list 'package-archives '("org" . "https://orgmode.org/elpa/") t)
(package-initialize)
```

## 4.2 use-package

Ensure `use-package` is installed, as well as all packages described in this configuration file.

```
(unless (package-installed-p 'use-package)
  (package-refresh-contents)
  (package-install 'use-package)
  (eval-when-compile (require 'use-package)))
(setq use-package-always-ensure t)
(require 'use-package)
(require 'bind-key)
```

## 4.3 TODO Convenient package update

One-function rollup of upgradeable package tagging, download and lazy install.

## 4.4 ivy

Auto completion.

```
(use-package ivy
  :config
  (setq ivy-use-virtual-buffers t
        ivy-count-format "%d/%d "
        enable-recursive-minibuffers t))
(ivy-mode t)
```

### 4.4.1 counsel

Wonderful counsellor!



```
(use-package counsel
  :bind ("M-x" . counsel-M-x)
  :config (counsel-mode t))

(global-set-key (kbd "C-f") 'counsel-grep-or-swiper)
```

#### 4.4.2 swiper

```
(use-package swiper
  :bind (("C-f" . counsel-grep-or-swiper)))
```

#### 4.5 evil-mode

Forgive me, for I have sinned.

```
(use-package evil)
(setq evil-toggle-key "C-c d") ; devil...
(evil-mode 1)
```

#### 4.6 org-mode

Phew, I can finally introduce Org mode! I am so **excited**.

Org mode replaces a word processor, a presentation creator, and a spreadsheet editor. IMHO, the spreadsheet ability captures more than 80% use cases wherein one wishes to include a table in a text document destined for physical publication. (It is clear that Excel spreadsheets are *not* destined for physical publication—simply attempt to print an Excel spreadsheet with the default settings.) In my opinion, Org mode matches all *useful* features of the Microsoft Office suite 1-to-1.

What follows are customizations designed to make Org mode behave more like Microsoft Word. The end goal is, once again, to draw as many new users to Emacs as possible!

##### 4.6.1 Basic customization

Org base directory is in user home on GNU/Linux, or in AppData in MS Windows.

```
(setq org-directory (concat user-emacs-directory "~/org"))
```

First, we hide markup symbols for **bold**, *italic*, underlined and ~~striketrough~~ text, and ensure our document appears indented upon loading:<sup>3</sup>

For the time being, I will in fact display emphasis markers, because hiding them corrupts tables.

```
(setq org-hide-emphasis-markers nil)
(setq org-startup-indented t)
```

Then, we customize Org headings to emulate WYSIWYG<sup>4</sup> behavior normally found in Word:

#### 4.6.2 Invisible edits

```
(setq org-catch-invisible-edits t)
```

#### 4.6.3 Agenda

The agenda displays a chronological list of headings across all agenda files for which the heading or body contain a matching `org-time-stamp`.<sup>5</sup>

```
(global-set-key (kbd "C-c a") 'org-agenda-list)
```

#### 4.6.4 Timestamps

More literary timestamps are exported to L<sup>A</sup>T<sub>E</sub>X using the following custom format:

```
(setq org-time-stamp-custom-formats
      '("%d %b, %Y (%a)" . "%d %b, %Y (%a), at %H:%M"))
```

#### 4.6.5 L<sup>A</sup>T<sub>E</sub>X export

The following makes CLOSED items appear green in L<sup>A</sup>T<sub>E</sub>X. Very stylish, much flair!

```
(setq org-latex-inactive-timestamp-format
      "\\textcolor{ForestGreen!60}{\\textit{%s}}")
```

---

<sup>3</sup>It *appears* indented, but the underlying plaintext file does not contain tab characters!

<sup>4</sup>What You See Is What You Get (input and output are identical), as opposed to What You See Is What You Mean (the input contains instructions that can modify the output).

<sup>5</sup>An `org-time-stamp` can be inserted with C-c . (period)

#### 4.6.6 Publish

In the following *alist* (association list), we describe the projects publishable via `org-publish`. We separate the publishing of `.org` files and attachments, because an online tutorial recommended we do so.

```
(require 'ox-publish)
(setq org-publish-project-alist
'(
  ("Safran-VIP-html"
   :base-directory "~/org/WORK/Safran/programs/B787/VIP/doc/org/"
   :base-extension "org"
   :publishing-directory "~/org/WORK/Safran/programs/B787/VIP/doc/wiki/"
   :recursive t
   :publishing-function org-html-publish-to-html
   :auto-preamble t
   :auto-sitemap t
   :sitemap-title "" )

  ("Safran-VIP-static"
   :base-directory "~/org/WORK/Safran/programs/B787/VIP/doc/org/"
   :base-extension "css\\|js\\|png\\|jpg\\|gif\\|pdf\\|mp3\\|mp4\\|ogg\\|swf"
   :publishing-directory "~/org/WORK/Safran/programs/B787/VIP/doc/wiki/"
   :recursive t
   :publishing-function org-publish-attachment )

  ("Safran-VIP-all"
   :components ("Safran-VIP-html" "Safran-VIP-static"))

  ("Safran-MA700-html"
   :base-directory "~/org/WORK/Safran/programs/MA700/doc/org/"
   :base-extension "org"
   :publishing-directory "~/org/WORK/Safran/programs/MA700/doc/wiki/"
   :recursive t
   :publishing-function org-html-publish-to-html
   :auto-preamble t
   :auto-sitemap t
   :sitemap-title "" )

  ("Safran-MA700-static"
   :base-directory "~/org/WORK/Safran/programs/MA700/doc/org/"
```

```

:base-extension "css\\|js\\|png\\|jpg\\|gif\\|pdf\\|mp3\\|mp4\\|ogg\\|swf"
:publishing-directory "~/org/WORK/Safran/programs/MA700/doc/wiki/"
:recursive t
:publishing-function org-publish-attachment )

("Safran-MA700-all"
 :components ("Safran-MA700-html" "Safran-MA700-static"))))

(add-to-list 'org-latex-packages-alist '("table" "xcolor"
  t ("pdflatex"))))
(add-to-list 'org-latex-packages-alist '("AUTO" "babel"
  t ("pdflatex"))))
(add-to-list 'org-latex-packages-alist '("AUTO" "polyglossia"
  t ("xelatex" "lualatex"))))

```

#### 4.6.7 Export

This creates a shorter binding for the most common Org export: Org  $\rightarrow$  L<sup>A</sup>T<sub>E</sub>X  $\rightarrow$  PDF.

```

(defun blendoit-org-quick-export ()
  "Org export to PDF and open.
  This basically reimplements 'C-c C-e l o'."
  (interactive)
  (org-open-file (org-latex-export-to-pdf)))

(global-set-key (kbd "C-c e") 'blendoit-org-quick-export)

```

#### 4.7 undo-tree

```
(global-undo-tree-mode)
```

#### 4.8 dumb-jump

```

(use-package dumb-jump)
(add-hook 'xref-backend-functions #'dumb-jump-xref-activate)

```

#### 4.9 gnuplot

```
(use-package gnuplot)
```

## 4.10 Ledger

```
(use-package ledger-mode
  :bind
  ("C-c r" . ledger-report)
  ("C-c C" . ledger-mode-clean-buffer))
```

## 4.11 TODO Sidebar

Get inspiration from `ibuffer-sidebar` and create a better sidebar.

```
;; (load-file)
```

## 4.12 which-key

```
(use-package which-key
  :init
  (which-key-mode)
  ;; :config
  ;; (setq which-key-idle-delay 1000)
  ;; (setq which-key-idle-secondary-delay 0.05)
  ;; (setq which-key-show-early-on-C-h t)
  )
```

## 4.13 company

```
; (add-hook 'after-init-hook 'global-company-mode)
```

## 4.14 flycheck

```
(use-package flycheck
  :init (global-flycheck-mode))
```

## 4.15 csv-mode

```
(use-package csv-mode)
```

## 4.16 json-mode

Oí, Jason!

Not needed in 27.1?

```
; (use-package json-mode)
```

#### 4.17 magit

```
(use-package magit
  :bind ("C-c g" . magit-status))
```

#### 4.18 pdf-tools

```
(use-package pdf-tools)
;; (pdf-tools-install)
```

#### 4.19 dashboard

We replace the standard welcome screen with our own.

```
(setq inhibit-startup-message t)
(use-package dashboard
  :config
  (dashboard-setup-startup-hook)
  (setq dashboard-startup-banner (concat user-emacs-directory "img/Safran_logo.svg"))
  (setq dashboard-items '((recents . 5)
    (projects . 5)))
  (setq dashboard-banner-logo-title "A modern professional text editor."))
```

#### 4.20 rainbow-mode

This highlights hexadecimal numbers which look like colors, in that same color.

```
(use-package rainbow-mode
  :ensure t
  :init
  (add-hook 'prog-mode-hook 'rainbow-mode))
```

#### 4.21 projectile

This enables us to better manage our .git projects.

```
(use-package projectile
  :bind ("C-c p" . 'projectile-command-map)
  :init (projectile-mode 1)
  (setq projectile-completion-system 'ivy))
```

#### 4.22 all-the-icons

```
(use-package all-the-icons)
```

#### 4.23 Better parentheses

```
(use-package rainbow-delimiters
  :config (add-hook 'prog-mode-hook #'rainbow-delimiters-mode))
(electric-pair-mode)
```

### 5 Cosmetics

#### 5.1 Cursors

In order to imitate other modern text editors, we resort to a blinking bar cursor. We choose red, the most captivating color, because the cursor is arguably the region on our screen:

1. most often looked at;
2. most often searched when lost.

##### 5.1.1 Default cursor

In files containing only **fixed-pitch** fonts (i.e. files containing only code), the cursor becomes a high-visibility box.

In files containing a mix of **variable-pitch** and **fixed-pitch** fonts, the cursor is a more MS Word-like bar.

```
(setq-default cursor-type (quote box))
(setq-default mixed-pitch-variable-pitch-cursor (quote bar))
```

#### 5.2 Faces

- **default**: Hack
  - Legible, modern monospace font
  - Strict, sharp, uncompromising
- **fixed-pitch**: Hack
- **variable-pitch**: Liberation Sans

- Libre alternative to Arial
- **org-block:** Hermit
  - Slightly wider than Hack
  - More opinionated shapes
  - Very legible parentheses

### 5.2.1 variable-pitch-mode

We use `variable-pitch-mode` for appropriate modes.

```
(add-hook 'org-mode-hook 'variable-pitch-mode)
(add-hook 'info-mode-hook 'variable-pitch-mode)
```

### 5.2.2 TODO Default font size

Make default font size larger on displays of which the resolution is greater than 1920x1080.

## 5.3 Initial and default frames

We set the dimensions of the initial and default frames.

```
(add-to-list 'default-frame-alist '(width . 80))
(add-to-list 'default-frame-alist '(height . 32))

(add-to-list 'initial-frame-alist '(width . 80))
(add-to-list 'initial-frame-alist '(height . 32))
```

### 5.3.1 GNU/Linux

These settings affect the first and subsequent frames spawned by Emacs in GNU/Linux. Frame transparency increases when focus is lost.

```
(when (and (display-graphic-p) (string-equal system-type "gnu/linux"))
  (set-frame-parameter (selected-frame) 'alpha '(90 . 50))
  (add-to-list 'default-frame-alist '(alpha . (90 . 50))))
```



## 5.4 Theme

My custom themes.

```
(setq custom-theme-directory (concat user-emacs-directory "themes/"))  
(load-theme 'blendoit-light)  
; (load-theme 'blendoit-dark)
```

### 5.4.1 blendoit-light

A highly legible and unambiguous theme.

The default face is a black foreground on a white background, this matches with MS Word.

**Red** TODO; cursor; negative; danger

**Green** DONE; positive

**Blue** document structure

**Purple** org-mode elements

### 5.4.2 blendoit-dark

Same principle

## 6 Editing preferences

These customizations enhance editor usability.

A line of text is considered ‘filled’ when it reaches 79 characters in length.

```
(setq-default fill-column 79)
```

We replace the longer **yes-or-no-p** questions with more convenient **y-or-n-p**.

```
(defalias 'yes-or-no-p 'y-or-n-p)
```

Disable minibuffer scroll bar.

```
(set-window-scroll-bars (minibuffer-window) nil nil)
```

Save cursor location in visited buffer after closing it or Emacs.

```
(save-place-mode 1)
```

Saving any file in `user-emacs-directory` (by default on Emacs, `~/emacs.d`) shall byte-recompile the entire `.emacs/` directory, for increased speed.

```
;; (defun my/byte-compile-user-config ()
;;   "Byte-compile dotfiles if current file is in 'user-emacs-directory'.
;;   Also tangles 'my-literate-config'."
;;   (interactive)
;;   (if (string-equal buffer-file-name my/literate-config)
;;       (org-babel-tangle))
;;   (if (string-prefix-p user-emacs-directory default-directory)
;;       (byte-recompile-directory (concat user-emacs-directory "blendoit/") 0)))

;; (add-hook 'after-save-hook my/byte-compile-user-config)
```

## 6.1 Clean up menus

Originally, I wished to inhibit certain entries in the GUI menus. Not worth the effort at this time.

```
(setq menu-bar-mode -1)
(setq tool-bar-mode -1)
```

## 6.2 Coding standards

This is just a better default. Don't @ me.

```
(setq c-default-style "linux"
      c-basic-offset 4)
```

## 6.3 Dividers

This ensures users can resize windows using the GUI.

```
(menu-bar-bottom-and-right-window-divider)
```

## 6.4 Tabs

### 6.5 auto-fill

Automatically break lines longer than `fill-column`.

```
(add-hook 'org-mode-hook 'turn-on-auto-fill)
```

## 6.6 Recent files

```
(recentf-mode 1)
(setq recentf-max-menu-items 25)
(setq recentf-max-saved-items 25)
(run-at-time nil (* 5 60) 'recentf-save-list)
```

## 6.7 pop-up-frames

```
; (setq pop-up-frames (quote graphic-only))
```

## 7 Late setup

### 7.1 Profiling — stop

```
;; (profiler-stop)
```

### 7.2 Profiling — report

```
;; (profiler-report)
```

## 8 Conclusion

In this configuration file, we described a series of customization steps taken to make Emacs more palatable to modern IDE users.