Emacs Turbo-Charges My Writing

Jeremy Friesen

[2023-10-03 Tue 09:45]

Contents

Emacs Turbo Charges My Writing

Presenter Jeremy Friesen

Pronouns he/him

Event URL `https://emacsconf.org/2023/talks/writing/`

Audience EmacsConf 2023.

Presentation Date and Time December 2, 2023 (2023-12-02) at 1:00 Post Meridian; after noon (pm) Eastern Daylight Savings Time (EDT).

Subjects Emacs, Writing

Socials **Blog** Take on Rules (takeonrules.com)

Mastodon dice.camp/@takeonrules

Email jeremy@jeremyfriesen.com

Introduction

I've been employed as a software developer since 1998 (1998). I've been writing for my personal blog since 2011 (2011). In May of 2020 (2020-05) I switched to Emacs; Previously **Atom!** (**Atom!**), Sublime Text, and Textmate; with numerous failed forays into Vim and a turn away in disgust of Visual Studio Code: Microsoft Sponsored Text Editor (VS Code) and it's belittling treatment of real estate for writing commit messages.

Curious about my public writing habits, I added a Post Statistics page to my personal site. Prior to Emacs, I had about a 95 words per day public output. After switching to Emacs I'm now averaging 340 words per day.

Delta, Delta, Delta

There are many reasons for this change:

- A pandemic removed my commute of 2 hours per day.
- A pandemic shifted my habits towards more alone/contemplative time.
- My children graduated secondary school and moved out.
- My wife's business became more self-sustaining.
- I expanded the scope of my blog; from gaming towards everything that I want to publicly write about.
- I changed employers; to one that encouraged public knowledge sharing.

All of this with Emacs as my supporting technology.

I have extended *my* Emacs to reduce barriers to capturing, synthesizing, referencing, and publishing my thinking and knowledge sharing.

And I'm always looking for refinements.

Well, How Did I Get Here?

I switched from WordPress to Jekyll then to Hugo. When I switched from WordPress to Jekyll I fully adopted Markdown.

During that switch I was learning Emacs; and hesitant to learn Org-Mode. It seemed like too much to introduce.

I was still keeping my public writing separate from my personal/private writing.

Keep Em' Separated

I didn't realize how much that separation created mental friction in my writing:

- Where should I write this?
- If I want to switch from private to public, what might I need to sanitize?

Further complicating things, private stuff used one markup format; public stuff had different markup.

I wrote down a data flow diagram to help think through how all local writing and links could move to a world-viewable state without private information also bleeding out.

Put another way, I have done the "pre-thinking" about where things should go. Now I file them and can easily refile them.

- Domains for Notes

  For this presentation, there are four notable domains:

  blog-post I'm going to be publishing this.

  epigraph Meaningful quotes that I've gathered in my writing.

  glossary These are a term or concept.

  melange I don't know where it goes, so it goes here.

  It is trivial to move notes between domains.

Exploratory Exhumation and Exhilarating Exposition

I began exploring Org-Mode, via Org-Roam, while also reading How to Take Smart Notes by Sönke Ahrens.

I experimented with some of those Zettelkasten approaches and began a journey of crafting and continuously refactoring my Personal Knowledge Management (PKM) process. What comes next is a demonstration.

In that time I've switched from the amazing Org-Roam (by Jehtro Kuan) to the simplified (see less dependencies) Denote package (by Protesilaos Stavrou (Prot)).

Demo Time

A stark difference for my preferred environment; be it writing or coding is that I don't use:

- tabs
- file/directory tree navigator
- multiple monitors

I aim to have one visible buffer; and split as needed. In other words, my hope is to minimize focal distractions.

Let's get started.

Create a Note

I have mapped the `H-d c` prefix to the mnemonic "Denote Create…" something. Through the which key package, I then show the functions for each of the "domains" I might write about.

Let's start a blog post.[1]

Dabbrev and Hippie Expand

After watching Jay Dixit's Emacs for Writers presentation, I adopted Dabbrev for auto-correction. I also make use of Hippie expand for word completion.

While pairing with those that use VS Code, I always wonder what tools they could use to match the simple functionality of the Hippie Expand and Dabbrev packages.

These are now so foundational, I don't think about them. They just make writing easier.

Links

Let's make a few links:

- Let's link to a note that does not have a "public" representation.
- Let's link to a note that references a "public" resource (Example given (e.g.) has a Universal Resource Locator (URL))
  - These will export as `CITE` and/or `A` tags.
- Let's convert a note link from a "title" to an abbreviation.
  - Abbreviations export as `ABBR` tags.
- I also have a `date:` type link
  - Exports as `TIME` tag.
- Let's insert an epigraph.
- Back-links

  I was at a conference and Jonathan Rochkind shared links with me that are very relevant to a project I've been working on. I chose to

[1] My blog makes extensive use of sidenotes and marginalia.

write the relevant information on my glossary entry. I'm relying on my back-linking to find that relevant information from Jonathan.

I want my notes to move towards the most relevant note, but be discoverable based on how my brain filed things away. With back-linking, I'm regularly leaving breadcrumbs to help find my way to the relevant information.

I haven't written the back-link functionality for `date:` Org-Mode functionality, but it's in my future consideration.

- Moving Through the Thoughts

  Put another way, links and back-links provide another means of discovery within my 3400 or so notes.

  As are hand-crafted indices; file searches; file names; and tags.

Clocking

Sometimes, I'll be writing and want to reference code (or prior blog posts). I'll use Org-Mode's capture to clock functionality to help quickly gather references. There are other ways to approach this, but it's what I have.

Macros

Hyper Text Markup Language (HTML)'s `I`-tag is semantically different from `EM`-tag; which is semantically different from the `CITE`-tag. All of which, by default, render as oblique or italics.

I use Daniel Mendler's Tempel package. For Org-Mode, I have a few templates that inject macro markup.

When I cite non-glossary entries, I use `H-m c` to insert a `cite` macro, and prompt for existing "cite" references in the document. Or `C-u H-m c` to prompt for all sources in my PKM.

I have `cite`, `keyboard`, `idiomatic`, and `mechanic` that are all part of the `H-m` prefix group. These macros evaluate differently for Hugo compared to LaTeX.

These macros help me write more semantic HTML.

Blocks

Org-Mode's default `org-insert-structure-block` is great, but I want further prompts. So I wrote a function that will take typing ‘‘‘ as the first characters of a line and prompt for the type of block.

Let's look at the "Blockquote" option. It prompts me for author, cite, and cite url. It builds from the `org-insert-structure-block` but prompts me to consistently fill out optional information.

These prompts also help me remember the syntax of writing these custom blocks.

Abstract

As I wrap up my writing, I include an abstract/summary. This is my chance for synthesis and to help possible future readers determine if this article is relevant to their interests.

It's also what I use for the page's meta description and eventually text for posting the link to Mastodon.

When I first started blogging, I wanted to "publish" and didn't take the time to summarize. Now, I consider it useful to "pitch" the document I wrote.

Export

With all of that, I export the document to my blog (`https://takeonrules.com`).

Let's look at the Markdown.

And the home page: `http://localhost:1313`

Note: I have a Hugo server running on this port.

Conclusion

When I started learning Emacs, I quickly shifted to "Vanilla" Emacs and just started writing.

As I wrote, when I needed to do something that I'd previously done in a text editor, I'd find and experiment with different packages.

I continue with that mindset. As I write, I'm attending to what I'm doing. And eventually I realize "if I were to just write a function that does this one thing... I'd have a smoother writing experience."

The goal is to minimize focal distractions, while also helping me create breadcrumbs that lead back to my writing and thoughts.

One of those "functions" I'd like to write is extending my `abbr:` export to work with LaTeX; it's halfway there I just need to prioritize and spend an evening of playing with Emacs.

Prologue

My Emacs configuration is at `https://github.com/jeremyf/dotemacs`

My blog is at `http://takeonrules.com`

Feel free to contact me:

Via Email jeremy@jeremyfriesen.com

Mastodon `http://dice.camp/@takeonrules`

Abbreviations

e.g. Example given

HTML Hyper Text Markup Language

EDT Eastern Daylight Savings Time

pm Post Meridian; after noon

PKM Personal Knowledge Management

VS Code Visual Studio Code: Microsoft Sponsored Text Editor

Prot Protesilaos Stavrou

URL Universal Resource Locator