Text Editors for Work

Adrian Kosmaczewski

2022-01-21

There has been a particular text editor that defined each period of my career as a software developer. This article is a summary of that history, so far.

I'm not mentioning the various IDEs that I had to use in the same points in time, because they are really specific to a particular technology: Visual Studio for C#, Xcode for Objective-C and Swift, etc. This is only about small, fast, flexible tools that have helped me write prose and code throughout the years.

If you like this, I've written more about the subject in one of my books (EPUB and HTML version).

- EditPlus (1998–2007)
- Smultron (2003–2006)
- TextMate (2006–2011)
- MacVim (2011-2016)
- vim (2002–now)
- Visual Studio Code (2016–now)
- Emacs (2018-now)

EditPlus (1998–2007)

I loved EditPlus since the first day I opened it, and it was my personal companion at writing code for years. Small, incredibly fast, with plenty of syntax files contributed by users. I used it to write all of my HTML, CSS, Active Server Pages, VBScript code, and more with it. As I stopped working on Windows, I stopped using it. I'm glad it is still around after all these years, though.

Smultron (2003–2006)

I bought an iBook G3 in December 2002 to learn Unix and Objective-C; but for a couple of months I struggled to find a text editor as good as EditPlus. The first one that actually worked for me was Smultron, a rather obscure text editor from Sweden, by far not the most popular in the platform, but extremely good at what it did.

TextMate (2006–2011)

TextMate was the editor that David Heinemeier Hansson used to demo Ruby on Rails. We can still watch this demo in what is now one of the oldest videos in YouTube. The success of Ruby on Rails propulsed TextMate to the top; however, the lack of progress towards TextMate 2.0 made me look elsewhere.

MacVim (2011-2016)

I used MacVim for a few years on the Mac, as my default GUI text editor. It had pretty good integration with MacOS, and it used the same shortcuts and plugins as the standard vim, so it was perfect in every way.

vim (2002–now)

As I said, one of the reasons I bought my iBook G3 in 2002 was to learn the Unix command line. And that's how I started using vim every day. I started piling up customizations in my `.vimrc`, then started using the Janus distribution, and finally settled for just a few selected plugins installed with Pathogen. Almost 20 years after, I think learning vim was one of the best investments of time. I use it in every computer I work with.

Visual Studio Code (2016–now)

I started using Visual Studio Code in 2018, as fate took me to work in a few projects based on TypeScript. I loved it, and these days I use it for anything that does not happen on a command line. I've written an article about the myriad of extensions that I use with it. It has great support for pretty much anything that I do, and it is cross-platform, which means that as I moved to Linux, it came with me.

Emacs (2018-now)

I had never used Emacs before 2018, and I wrote a book as I learnt to use it, mostly comparing it with vim. I like Emacs a lot, both in GUI and TUI mode, and in particular I appreciate its integration with Git using Magit and its Asciidoc mode.