

# **A Memo About Memos**

Alejandro Revilla (@apr)

2024-07-24

## Contents

<b>A Memo About Memos</b>	<b>2</b>
Required tools . . . . .	3
TexLive . . . . .	3
Pandoc . . . . .	3
Mermaid-filter . . . . .	3
M4 . . . . .	3
Pandoc template . . . . .	4
The memo script . . . . .	4
Frontmatter . . . . .	7
Bibliography . . . . .	7
Revision History . . . . .	8
<b>References</b>	<b>8</b>

## A Memo About Memos

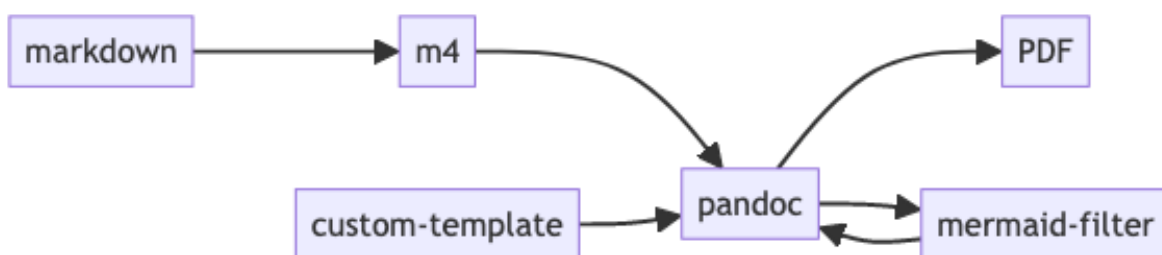
I am such a weak writer that I can't afford to add bad typography on top of my already flawed writing. I'm so challenged when it comes to aesthetics that, whenever I use a WYSIWYG editor—and I've used many since the early PageMaker days—I spend more time figuring out fonts and paragraph alignment than actually writing. Because I occasionally get to read beautifully typeset documents—and terrible ones most of the time—I can tell the difference and appreciate subtle details such as appropriate hyphenation, justification, kerning, leading, and the use of ligatures.

I write a lot of memos for different audiences, including technical, executive, internal, customer proposals, design analysis, and more. Over the years, I have used various tools to write these memos, such as YODL [1],  $\text{T}_{\text{E}}\text{X}$ [2],  $\text{\LaTeX}$ [3], DocBook [4], DITA [5], AsciiDoc [6], AsciiDoctor [7], and of course, Markdown<sup>1</sup>. After decades of peregrination through different toolchains and editors, including high-end XML editors, nothing beats `vi` (or `emacs` if that's your religion) for fast editing, `Markdown` for ease of use, and  $\text{\LaTeX}$  for perfect typesetting.

While the awesome `pandoc` is the easiest way to convert `Markdown` to a properly typeset `PDF`, the default output, though better than that of many other Markdown editors, is great for occasional memos but doesn't look professional. I've been trying to write a nice `pandoc` template myself without much success, but I recently came across a beautiful one: Eisvogel.<sup>2</sup>

This document could end right here: write Markdown and use 'pandoc' with the 'Eisvogel' template to produce beautiful documents. Period.

However, I believe I can contribute a small addition to the toolchain that will enhance your writing experience and content reuse.



<sup>1</sup> Disregard the bibliographic references and bookmarks like this; I'm adding them just to show how they look.

<sup>2</sup> <https://github.com/Wandmalfarbe/pandoc-latex-template>

## Required tools

### TexLive

T<sub>E</sub>XLive is a comprehensive distribution of the T<sub>E</sub>X typesetting system, offering a wide range of T<sub>E</sub>X-related software, packages, fonts, and utilities. Maintained by the T<sub>E</sub>X Users Group, it supports multiple operating systems, including Linux, macOS, and Windows. The live distribution can be found at: <https://www.tug.org/texlive/>

### Pandoc

Pandoc is a powerful document converter that supports over a dozen input formats and more than thirty output formats, including Markdown, HTML, L<sup>A</sup>T<sub>E</sub>X, PDF, and Word, making it a perfect fit for our memo toolchain and can be found at <https://pandoc.org/>

### Mermaid-filter

Mermaid is a versatile tool for creating diagrams and charts from simple text descriptions. It generates flowcharts, sequence diagrams, Gantt charts, and more using an easy-to-learn syntax. Popular in technical documentation and project management, Mermaid integrates seamlessly with various platforms, enhancing documents with clear and visually appealing diagrams. We need the `mermaid-filter` package that gets called from `pandoc`, and can be installed with the command:

```
npm i -g mermaid-filter
```

### M4

The `m4` macro processor, designed by Brian Kernighan and Dennis Ritchie and originally developed as part of the Unix operating system in the 1970s, is not technically required in this toolchain but is a perfect fit for splitting large documents into smaller pieces, reusing content snippets, and providing standard constants such as the Git revision of the document, date, time, filename, and more. Key features of `m4` include its ability to define macros, perform arithmetic operations, manipulate text with ease, and call the operating system<sup>3</sup>. It also supports file inclusion, conditional statements, and looping constructs.

As an example, to show the revision of the document, you can insert `__REVISION__`, which will be expanded in this case to `ddaa199*`. The same goes for other variables such as `__DATE__`, `__TIME__`, `__FILENAME__`, `__BRANCH__` and others that you may want to add to the `memo`

---

<sup>3</sup> This poses a security risk when using third-party Markdown sources.

script. If you're on Unix, `m4` is probably installed, verify that with the command `type m4`. On Windows, you can install it using Cygwin, or the Windows Subsystem for Linux (WSL).

## Pandoc template

The Eisvogel template, located at <https://github.com/Wandmalfarbe/pandoc-latex-template>, has a fantastic set of examples and provides detailed instructions on how to install it. On macOS, with `pandoc` installed via `brew`, the location is `$HOME/.local/share/pandoc/templates` under the name `custom_eisvogel.latex`. It can, of course, be the pristine `eisvogel.latex` script. I suggest calling it `custom_eisvogel.latex` in case you want to add some additional  $\text{\LaTeX}$  commands or configuration.

In that `pandoc/templates` directory, you may also want to add `ieee-with-url.csl` from <https://github.com/citation-style-language/styles/blob/master/ieee-with-url.csl> or another citation style of your choice.

## The memo script

```
1  #!/bin/bash
2
3  # Requires npm i -g mermaid-filter and TeXLive
4
5  set -e
6
7  # print usage
8  usage() {
9      echo "Usage: $(basename $0) [-DNAME=VALUE ...] your-markdown-file.
10         md [single|double]"
11      exit 1
12  }
13
14  # initialize Git branch and revision
15  init_git_info() {
16      BRANCH="unknown"
17      REVISION="unknown"
18
19      if git rev-parse --is-inside-work-tree > /dev/null 2>&1; then
20          if git rev-parse HEAD > /dev/null 2>&1; then
21              BRANCH=$(git rev-parse --abbrev-ref HEAD)
22              REVISION=$(git rev-parse --short=7 HEAD)
23              if [ -n "$(git status --porcelain)" ]; then
24                  REVISION="${REVISION}*"
25              fi
26          else
27              BRANCH="empty"
28              REVISION="empty"
29          fi
30      fi
31  }
```

```
29     fi
30 }
31
32 # parse layout option
33 parse_layout_option() {
34     if [ "$1" = "single" ]; then
35         LAYOUT_OPTION=""
36     elif [ "$1" = "double" ]; then
37         LAYOUT_OPTION="--variable classoption=twocolumn"
38     else
39         echo "Invalid layout option. Use 'single' or 'double'."
40         exit 1
41     fi
42 }
43
44 # Check for minimum number of arguments
45 if [ $# -lt 1 ]; then
46     usage
47 fi
48
49 # Initialize variables for m4 -D parameters
50 TS=$(date "+%Y-%m-%d %H:%M" | awk '{ printf "%s", $0 }')
51 D=$(echo "$TS" | cut -d ' ' -f 1)
52 T=$(echo "$TS" | cut -d ' ' -f 2)
53
54 # Initialize Git branch and revision
55 init_git_info
56
57 # Initialize m4 parameters
58 M4_PARAMS="-D__DATE__=$D -D__TIME__=$T -D__BRANCH__=\"$BRANCH\" -
D__REVISION__=$REVISION"
59 PANDOC_TEMPLATES_DIR="$HOME/.local/share/pandoc/templates"
60
61 # Parse -D parameters and Markdown file
62 while [ $# -gt 0 ]; do
63     case "$1" in
64         -D*)
65             M4_PARAMS="$M4_PARAMS $1"
66             shift
67             ;;
68         *.md)
69             MD=$1
70             shift
71             break
72             ;;
73         *)
74             usage
75             ;;
76     esac
77 done
78
79 # Check if a Markdown file was provided
80 if [ -z "$MD" ]; then
```

```
81     usage
82 fi
83
84 # Parse layout option
85 LAYOUT=${1:-single}
86 parse_layout_option "$LAYOUT"
87
88 DIR=$(dirname "$MD")
89 BASENAME=$(basename "$MD" .md)
90 PDF="$DIR/${BASENAME}.pdf"
91 BIB="$DIR/${BASENAME}.bib"
92 PREPROCESSED_MD="$${BASENAME}_preprocessed.md"
93 ERROR_FILE="mermaid-filter.err"
94 M4_PARAMS="$M4_PARAMS -D__FILENAME__=$MD"
95
96 # Preprocess the Markdown file with m4
97 {
98     echo ""
99     echo ""
100    cat "$MD"
101 } | m4 $M4_PARAMS - > "$PREPROCESSED_MD"
102
103 # Use -t latex -o $MD.tex to get LaTeX output instead of PDF
104 if [ -f "$BIB" ]; then
105     if [ -z "$BIB_OPTION" ]; then
106         BIB_OPTION="--bibliography=$BIB"
107     else
108         BIB_OPTION="$BIB_OPTION --bibliography=$BIB"
109     fi
110     BIB_OPTION="$BIB_OPTION --citeproc"
111 fi
112
113 cd "$DIR" && pandoc --pdf-engine=xelatex \
114     --from markdown \
115     --variable papersize=a4 \
116     --template=custom_eisvogel --listings \
117     --csl=$PANDOC_TEMPLATES_DIR/ieee-with-url.csl \
118     -F mermaid-filter \
119     $LAYOUT_OPTION \
120     $BIB_OPTION \
121     -s "$PREPROCESSED_MD" -o "$PDF"
122
123 # Clean up the preprocessed file
124 if [ -f "$PREPROCESSED_MD" ]; then
125     rm "$PREPROCESSED_MD"
126 fi
127
128 # Clean up mermaid's error file if it's empty
129 if [ -f "$ERROR_FILE" ] && [ ! -s "$ERROR_FILE" ]; then
130     rm "$ERROR_FILE"
131 fi
132
133 echo "PDF generated at $PDF"
```

```
134 open "$PDF"
```

## Frontmatter

This is a sample frontmatter for this document:

```
1 ---
2 title: A Memo About Memos
3 subject: Memo Typesetting
4 author: Alejandro Revilla (@apr)
5 date: 2024-07-24
6 keywords: [Typography]
7 titlepage: true
8 # titlepage-logo: jpos.jpg
9 lang: "en"
10 toc: true
11 toc-own-page: true
12 footer-center: jPOS.org
13 titlepage-rule-color: "360049"
14 titlepage-text-color: "FFFFFF"
15 titlepage-rule-color: "360049"
16 titlepage-rule-height: 0
17 titlepage-background: "../backgrounds/background.pdf"
18 footnotes-pretty: true
19 header-includes:
20   - \usepackage{multicol}
21   - \usepackage{mdframed}
22   - \usepackage{graphicx}
23   - \usepackage{xcolor}
24   - \usepackage{pgfplots}
25   - \usepackage{amsmath}
26   - \pgfplotsset{compat=1.17}
27 ---
```

Note that some of the added packages are used in my custom template, just as a PoC of different features. You can safely remove that `header-includes` section.

## Bibliography

You can add `.bib` file, with the same name as your main `.md` file with the following format:

```
1 @article{kubat1997yodl,
2   title = {YODL or Yet Oneother Document Language},
3   author = {Kubat, Karel},
4   journal = {Linux Journal},
5   year = {1997},
6   month = {November},
7   number = {2089},
8   url = {https://www.linuxjournal.com/article/2089}
```



```
9  }
10
11  @manual{docbook,
12    title = {DocBook 5.0: The Transition Guide},
13    author = {{OASIS DocBook Technical Committee}},
14    year = {2010},
15    note = {Available online at \url{https://www.docbook.org/tdg5/en/html/
16            /docbook.html}},
17    url = {https://www.docbook.org/tdg5/en/html/docbook.html}
18  }
19  @manual{dita,
20    title = {DITA 1.3 Specification},
21    author = {{OASIS DITA Technical Committee}},
22    year = {2015},
23    note = {Available online at \url{https://docs.oasis-open.org/dita/
24            dita/v1.3/os/}},
25    url = {https://docs.oasis-open.org/dita/dita/v1.3/os/}
26  }
27  ...
```

## Revision History

- 2024-11-01 (100cdf4) add revision history and better handling of dirty repos
- 2024-08-29 (748a83f) A memo about memos

## References

- [1] K. Kubat, “YODL or yet oneother document language,” *Linux Journal*, no. 2089, 1997, Available: <https://www.linuxjournal.com/article/2089>
- [2] D. E. Knuth, *The TeXbook*. in Computers & typesetting, volume a. Reading, Massachusetts: Addison-Wesley, 1984.
- [3] L. Lamport, *LaTeX: A document preparation system*, 2nd ed. Reading, Massachusetts: Addison-Wesley, 1994.
- [4] OASIS DocBook Technical Committee, *DocBook 5.0: The transition guide*. 2010. Available: <https://www.docbook.org/tdg5/en/html/docbook.html>
- [5] OASIS DITA Technical Committee, *DITA 1.3 specification*. 2015. Available: <https://docs.oasis-open.org/dita/dita/v1.3/os/>
- [6] E. Stuart, *AsciiDoc user guide*. 2002. Available: <http://www.methods.co.nz/asciidoc/>
- [7] D. Allen, *Asciidoctor documentation*. 2021. Available: <https://asciidoctor.org/docs/>