

# The c13n System Backend Introduction

rne

Sep 20, 2024

## 1 External Dependencies

### 1.1 md2tex utility

1. clang, gcc or any working c compiler.
2. gnu make.
3. glibc, musl, or any c standard library.

### 1.2 drv.ltx driver

1. LuaTeX.
2. LuaTeX-ja and evangelion-jfm.
3. Macro-package float, xurl, listings, graphicx, lua-ul, fontspec, micortype, ragged2e, hyperref and luatexja.

All of these above are contained in a standard full TeXLive (or MacTeX) installation that is released after 2024.

### 1.3 make.py build system

1. Python3 (tested on 3.9.6).

## 2 About the md2tex utility

### 2.1 Features

This is a Markdown to (La)TeX parser implemented in C, based on MD4C, with the following features:

1. **Compliance:** It is compliant to the latest version of CommonMark specification thanks to MD4C. Currently, we supports CommonMark 0.31.
2. **Extensions:** It supports some widely used extensions, namely: table, strikethrough,

underline, and TeX style equations.

3. **Lenience:** It follows completely the GIGO philosophy (garbage in, garbage out). It sees any sequence of bytes as valid input. It will not throw any error when parsing.
4. **Performance:** It is very fast, parsing most of our posts in less than 1 ms.
5. **Portability:** It is tested to run on Mach-O, iOS, and Linux. It should run on Windows, BSDs, and all other POSIX-compliant OSes which has a working C compiler.
6. **Encoding:** It understands UTF-8 to determine word boundaries, and case-insensitive matching of a link reference label (Unicode case folding). However, it will not translate HTML entities and numeric character references.

## 2.2 Build Instructions

### 2.2.1 POSIX-compliant OSes

Make sure you have a working C compiler (Clang/GCC/...) and standard library (GLibC/musl/...), together with GNU Make and sed tools.

Then simply run make to generate the executable md2tex.

### 2.2.2 Windows

You can install Cygwin and follow the same step, or you can use the fragile way.

```
1 cc -o md2tex md2tex.c md4c.c -O2 -Wall
```

## 2.3 Markdown Spec

Generally, you should conform to the CommonMark spec when writing posts. However, because of the additional extensions, there are some extra rules.

### 2.3.1 Table

It supports GitHub-style tables .

Note that in the generated (La)TeX manuscript, the align information of the columns are currently ignored, and default to left align. It may be supported in the future.

### 2.3.2 Strikethrough

It supports `~delete~` (delete).

### 2.3.3 Underline

Underscore denotes an underline instead of an ordinary emphasis or strong emphasis. Thus, to get italics and **bold**, use `*` instead.

### 2.3.4 TeX-style Equation

TeX like inline math spans (`$...$`) and display math spans (`$$...$$`) are supported. You are not required to escape ordinary dollars signs in most cases.

```

1 This is an inline math span: $a + b = c$.
3 This is a display math span:
  $$
5   a + b = c.
  $$
7
9 This is a dollar sign: $, 12$; equivalent to \$, 12\$.
11 This is a double dollar sign: $$; equivalent to \$\$.
13 Common knowledge: math spans cannot be nested.
   $$foo $bar$ baz$$ is equivalent to \$$foo $bar$ baz\$$ which only bar is rendered
   ↪ as equation.
15 Note: the opening delimiter or closing delimiter cannot be preceded or followed by an
   ↪ alphanumeric character.
   x$a + b = c$ or $a + b = c$y will not be rendered as math (all $ rendered as \$)

```

## 3 About the `drv[pst|mly].ltx` style sheet

It works only under LuaLaTeX. No plan for porting. The `pst` is for standalone posts while the `mly` is for monthly.

### 3.1 Text style

We use `\underline` and `\strikeThrough` to replace the LaTeX2e provided buggy `\underline` and the undefined `\del`.

### 3.2 Patch

Because SAX-like parser cannot elegantly capture the text between span delimiters, control sequence as listed below may appear in several circumstances.

```

1 \href{<url>}{} % from [](<url>)
2 \label{} % from an image that does not have a label

```

Instead of using further dark magic in the parser (i.e., modify the text callback), we handle this in LaTeX.

The control sequence `\href` is a hard one, because it relies on nasty catcode modifications to read in url containing chars that normally need to be escaped in a TeX manuscript, and thus means we cannot simply gobble the url into a parameter and further patch it. Also because of the complex infrastructure of `hyperref`, a custom `\href` has been implemented to solve this problem. It should have the exact functionality of `\href`, except when #2 is empty, we supply one which has value `\url{#1}`. This command is LuaTeX specific.

```

1 \def\inner@ifempty#1{\begingroup\toks0={#1}\edef\param{\the\toks0}%
2 \expandafter\endgroup\ifx\param\empty\expandafter\@firstoftwo\else%
  \expandafter\@secondoftwo\fi}
4 \def\inner@makeother#1{\catcode`#112\relax}
\def\href{\leavevmode\bgroup\let\do\inner@makeother\dospecials\inner@href}
6 \begingroup\catcode`=1\catcode`=2\catcode`\{=12\catcode`\}=12
  \gdef\inner@href{#1}{#2}[\pdfextension startlink user[/Subtype/Link/A<<Type/Action/S
    ↪ /URI/URI(#1)>>]\inner@ifempty{#2}[\url{#1}]{#2}\pdfextension endlink \egroup}\
    ↪ endgroup

```

Documenting this is unnecessary I think.

Without utilizing catcode dark magic, `\label` is a really easy one.

```

1 \begingroup\catcode`X=3\gdef\expnd@ifempty#1{%
  \ifX\detokenize{#1}X\expandafter\@firstoftwo\else%
3 \expandafter\@secondoftwo\fi}\endgroup
\let\furui@label\label
5 \def\label#1{\expnd@ifempty{#1}\relax\furui@label{#1}}

```

### 3.3 Headings

To make life easier for injecting metadata from `mdx`, also to enable sandboxed pdf generation for `monthly`, the `drvml]y.ltx` redefines some command.

```

1 \def\mlytitle#1{\title{#1}\author{c13n}\date{\today}\maketitle
  \gdef\title##1{\def\TITLE{##1\rlap{\quad\vtop{\normalsize\hbox{\AUTHOR}\
    ↪ \hbox{\DATE}}}}}}
3 \gdef\author##1{\xdef\AUTHOR{##1}}
  \gdef\date##1{\xdef\DATE{##1}}
5 \gdef\maketitle{\part{\TITLE}}

```

### 3.4 Blocks

We include the `float` package as every float uses `[H]`. As graphics are represented using the `\image` control sequence in the parser, we have the following definition.

```
1 \setkeys{Gin}{width=.75\csname Gin@width\endcsname,keepaspectratio}
  \def\image#1{\includegraphics{#1}}
```

Another rather simple one is the thematic break `\thematic`.

```
2 \newcommand{\thematic}{\vspace{2.5ex}\par\noindent%
  \parbox{\textwidth}{\centering[*]\[-4pt]{*}\enspace{*}\vspace{2ex}}\par}
```

### 3.5 CJK

We currently support 中文 using `luatexja`. To rebuild all posts, you only need a complete TeXLive (or MacTeX) installation as the appropriate font is distributed with this repo. Evangelion-JFM is used as the font metric.

### 3.6 LuaTeX

To silence the engine when batch compiling, some callback is modified:

```
1 \directlua{
2 function be_quiet () end
  luatexbase.add_to_callback('start_run', be_quiet, 'stop start run')
4 luatexbase.add_to_callback('stop_run', be_quiet, 'stop stop run')
  luatexbase.add_to_callback('start_page_number', be_quiet, 'stop start page')
6 luatexbase.add_to_callback('stop_page_number', be_quiet, 'stop stop page')
  luatexbase.add_to_callback('start_file', be_quiet, 'stop start file')
8 luatexbase.add_to_callback('stop_file', be_quiet, 'stop stop file')
  luatexbase.add_to_callback('show_warning_message', be_quiet, 'stop show warning
  ↪ message')
10 }
```

### 3.7 LaTeX

To the same reason, LaTeX (especially listings) is asked to keep silent:

```
1 \RequirePackage[immediate]{silence}
2 \WarningsOff
  \ErrorsOff[Listings]
```

## 4 The build system `make.py`

It's written in Python, and thus should be portable.

The only thing that should be mentioned about it is that please do not use `webp` as image format.

Use `python make.py post` to make posts (existing posts will not trigger rebuild) and `python make.py batch` to make all the monthly.

For more details, see the comments. This is the only file which is commented carefully.