

The latex-formatter CTAN package

v0.5.7

William G. Underwood

March 25, 2026

1 Overview

The latex-formatter CTAN package provides the tex-fmt command line tool for formatting LaTeX source files. Main features include:

- Extremely fast run-time performance
- Minimal configuration required
- Command line interface
- Handles LaTeX file types `.tex`, `.bib`, `.cls`, and `.sty`
- Written entirely in safe Rust

Before tex-fmt file.tex	After tex-fmt file.tex
<code>\documentclass{article}</code>	<code>\documentclass{article}</code>
<code>\begin{document}</code>	<code>\begin{document}</code>
<code>\begin{itemize}</code>	<code>\begin{itemize}</code>
<code>\item Lists with items</code>	<code>\item Lists with items</code>
<code>over multiple lines</code>	<code>over multiple lines</code>
<code>\end{itemize}</code>	<code>\end{itemize}</code>
<code>\begin{equation}</code>	<code>\begin{equation}</code>
<code>E = m c^2</code>	<code>E = m c^2</code>
<code>\end{equation}</code>	<code>\end{equation}</code>
<code>\end{document}</code>	<code>\end{document}</code>

2 Usage

The most commonly used options are given below. For a full list, see Section 4.

```
tex-fmt file.tex           # format file.tex and overwrite
tex-fmt --check file.tex  # check if file.tex is correctly formatted
tex-fmt --print file.tex  # format file.tex and print to stdout
tex-fmt --recursive      # recursively format files in
                          # current directory
tex-fmt --recursive dir/  # recursively format files in dir
tex-fmt --fail-on-change file.tex # format file.tex and return exit-code 1
                          # if overwritten
tex-fmt --nowrap file.tex # do not wrap long lines
tex-fmt --format-tables file.tex # format tables (align ampersands)
tex-fmt --stdin          # read from stdin and print to stdout
tex-fmt --help           # view help information
```

2.1 Configuration

Options can also be set using a configuration file, which will be read from the following locations, in order of decreasing priority.

- A named config file passed as `tex-fmt --config <PATH>`
- A file named `tex-fmt.toml` in the current working directory
- A file named `tex-fmt.toml` in the root directory of the current git repository
- A file named `tex-fmt.toml` in a subdirectory titled `tex-fmt/` in the user's configuration directory at
 - Linux: `~/.config/tex-fmt/tex-fmt.toml`
 - macOS: `/Users/<USER>/Library/Application Support/tex-fmt/tex-fmt.toml`
 - Windows: `C:\Users\<USER>\AppData\Roaming\tex-fmt\tex-fmt.toml`

Arguments passed on the command line will always override those specified in configuration files. To ignore all config files, use the `--noconfig` flag. An example configuration file is as follows.

```
# tex-fmt.toml
check = false
print = false
wrap = true
wraplen = 80
tabsize = 2
tabchar = "space"
stdin = false
verbosity = "warn"
lists = []
no-indent-envs = []
```

2.2 Disabling the formatter

Ending a source line with `% tex-fmt: skip` disables formatting for that line. To disable the formatter for a block, use `% tex-fmt: off` and `% tex-fmt: on`. Verbatim environments including `verbatim`, `Verbatim`, `lstlisting` and `minted` are automatically skipped.

```
\documentclass{article}
\begin{document}
  This line is skipped % tex-fmt: skip
% tex-fmt: off
  These lines are also
  not formatted or wrapped
% tex-fmt: on
\end{document}
```

2.3 Ignoring files in recursive mode

Recursive searches with `--recursive` or `-r` will ignore patterns in `.gitignore` and `.ignore` files, following git conventions.

2.4 Shell completion

Shell completion scripts can be generated at run-time using the `--completion <SHELL>` flag.

2.5 Man page

A man page can be generated at run-time using the `--man` flag.

3 Limitations

- Semantic parsing of LaTeX code not conducted
- No linting or correction of syntax errors
- Compliance with existing formatting guidelines not guaranteed
- No spelling or grammar checking

4 Options

4.1 Command line options

The following arguments can be passed on the command line.

Option	Alias	Default	Description
<code>--check</code>	<code>-c</code>		Check formatting, do not modify files
<code>--print</code>	<code>-p</code>		Print to stdout, do not modify files
<code>--fail-on-change</code>	<code>-f</code>		Fail if files are modified
<code>--recursive</code>	<code>-r</code>		Recursively search for files to format
<code>--nowrap</code>	<code>-n</code>		Do not wrap long lines
<code>--wraplen <N></code>	<code>-l</code>	80	Line length for wrapping
<code>--tabsize <N></code>	<code>-t</code>	2	Number of characters to use as tab size
<code>--usetabs</code>			Use tabs instead of spaces for indentation
<code>--format-tables</code>			Format tables
<code>--stdin</code>	<code>-s</code>		Process stdin as a single file, output to stdout
<code>--config <PATH></code>			Path to config file
<code>--noconfig</code>			Do not read any config file
<code>--verbose</code>	<code>-v</code>		Show info messages
<code>--quiet</code>	<code>-q</code>		Hide warning messages
<code>--trace</code>			Show trace messages
<code>--completion <SHELL></code>			Generate a shell completion script
<code>--man</code>			Generate a man page
<code>--args</code>			View arguments passed to tex-fmt
<code>--help</code>	<code>-h</code>		Print help
<code>--version</code>	<code>-V</code>		Print version

4.2 Configuration file options

The following arguments can be provided in `tex-fmt.toml`. The first example in each row is the default value.

Option	Type	Examples	Description
<code>check</code>	bool	<code>false</code>	Check formatting, do not modify files
<code>print</code>	bool	<code>false</code>	Print to stdout, do not modify files
<code>fail-on-change</code>	bool	<code>false</code>	Fail if files are modified
<code>wrap</code>	bool	<code>true</code>	Wrap long lines
<code>wraplen</code>	int	80, 100	Line length for wrapping
<code>wrapmin</code>	int	70, 90	Target minimum length for line wrapping
<code>tabsize</code>	int	2, 4	Number of characters to use as tab size
<code>tabchar</code>	str	<code>"space"</code> , <code>"tab"</code>	Character to use for indentation
<code>format-tables</code>	bool	<code>false</code>	Format tables
<code>stdin</code>	bool	<code>false</code>	Process stdin as a single file, output to stdout
<code>lists</code>	arr[str]	<code>[]</code> , <code>["myitemize"]</code>	Extra list environments to be formatted as <code>itemize</code>
<code>verbatim</code>	arr[str]	<code>[]</code> , <code>["myverbatim"]</code>	Extra verbatim environments
<code>no-indent-envs</code>	arr[str]	<code>[]</code> , <code>["mydocument"]</code>	Environments which are not indented
<code>wrap-chars</code>	arr[str]	<code>[]</code> , <code>[","]</code>	Characters after which lines may be wrapped
<code>verbosity</code>	str	<code>"warn"</code> , <code>"error"</code>	Verbosity level for terminal logging