The latex-formatter CTAN package
v0.5.6

William G. Underwood

October 20, 2025

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`

`\documentclass{article}`
 	
`\begin{document}`
 	
`\begin{itemize}`
`\item Lists with items`
`over multiple lines`
`\end{itemize}`
 	
`\begin{equation}`
`E = m c^2`
`\end{equation}`
 	
`\end{document}`
	`\documentclass{article}`
	
	`\begin{document}`
	
	`\begin{itemize}`
	`  \item Lists with items`
	`    over multiple lines`
	`\end{itemize}`
	
	`\begin{equation}`
	`  E = m c^2`
	`\end{equation}`
	
	`\end{document}`

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 --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
`--check`	`-c`		Check formatting, do not modify files
`--print`	`-p`		Print to stdout, do not modify files
`--fail-on-change`	`-f`		Fail if files are modified
`--recursive`	`-r`		Recursively search for files to format
`--nowrap`	`-n`		Do not wrap long lines
`--wraplen <N>`	`-l`	`80`	Line length for wrapping
`--tabsize <N>`	`-t`	`2`	Number of characters to use as tab size
`--usetabs`			Use tabs instead of spaces for indentation
`--stdin`	`-s`		Process stdin as a single file, output to stdout
`--config <PATH>`			Path to config file
`--noconfig`			Do not read any config file
`--verbose`	`-v`		Show info messages
`--quiet`	`-q`		Hide warning messages
`--trace`			Show trace messages
`--completion <SHELL>`			Generate a shell completion script
`--man`			Generate a man page
`--args`			View arguments passed to tex-fmt
`--help`	`-h`		Print help
`--version`	`-V`		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
`check`	bool	`false`	Check formatting, do not modify files
`print`	bool	`false`	Print to stdout, do not modify files
`fail-on-change`	bool	`false`	Fail if files are modified
`wrap`	bool	`true`	Wrap long lines
`wraplen`	int	`80`, `100`	Line length for wrapping
`wrapmin`	int	`70`, `90`	Target minimum length for line wrapping
`tabsize`	int	`2`, `4`	Number of characters to use as tab size
`tabchar`	str	`"space"`, `"tab"`	Character to use for indentation
`stdin`	bool	`false`	Process stdin as a single file, output to stdout
`lists`	arr[str]	`[]`, `["myitemize"]`	Extra list environments to be formatted as `itemize`
`verbatims`	arr[str]	`[]`, `["myverbatim"]`	Extra verbatim environments
`no-indent-envs`	arr[str]	`[]`, `["mydocument"]`	Environments which are not indented
`wrap-chars`	arr[str]	`[]`, `[","]`	Characters after which lines may be wrapped
`verbosity`	str	`"warn"`, `"error"`	Verbosity level for terminal logging