

NAME

texlogsieve – filter and summarize LaTeX log files

SYNOPSIS

texlogsieve [*OPTION*]... [*INPUT FILE*]

DESCRIPTION

texlogsieve reads a LaTeX log file (or the standard input if no file is specified), filters out less relevant messages, and displays a summary report.

texlogsieve **must** be run from the same directory as [pdf]lua[*xe*]latex, because it searches for the files used during compilation (packages loaded from the current directory, files included with \input etc.).

The program goes to great lengths to correctly handle TeX line wrapping. It understands the *max_print_line* TeX configuration variable and reads its value from the same places as TeX. Setting *max_print_line* to a value larger than 9999 makes texlogsieve ignore line wrapping.

The defaults are reasonable; hopefully, you can just do

```
[pdf|lua|xe]latex myfile.tex | texlogsieve
```

and be satisfied with the result.

OPTIONS

--page-delay, --no-page-delay

Enable/disable grouping messages by page before display. When enabled, messages are only output after the current page is finished (shipout). The advantage is that the page number is included in the message (default enabled).

--summary, --no-summary

Enable/disable final summary (default enabled).

--only-summary

No messages, show only the final summary (default disabled).

--shipouts, --no-shipouts

Enable/disable reporting shipouts (default disabled with page-delay, enabled with no-page-delay).

--repetitions, --no-repetitions

Allow/prevent repeated messages (default disabled, i.e., repeated messages are suppressed).

--heartbeat, --no-heartbeat

Enable/disable progress gauge in page-delay mode (default enabled).

-l LEVEL, --minlevel=LEVEL

Filter out messages with severity level lower than LEVEL. Valid levels are DEBUG (no filtering), INFO, WARNING, CRITICAL, and UNKNOWN (default WARNING).

-u, --unwrap-only

Do not filter messages and do not output the summary, only unwrap long, wrapped lines. The output should be very similar (but not equal) to the input file, but with wrapped lines reconstructed. This activates `-l debug`, `--no-summary`, `--no-page-delay`, `--repetitions`, and `--shipouts`, and also suppresses the verbose “open/close file” and “shipout” messages, simulating instead the TeX format, with parens and square brackets. This is useful if you prefer the reports generated by some other tool but want to benefit from texlogsieve’s line unwrapping algorithm; the output generated by this option should be parseable by other tools (but you probably need to coerce the other tool not to try to unwrap lines).

--silence-package=PKGNAME

Filter out messages that can be identified as coming from the given package. Use this option multiple times to suppress messages from several different packages.

--silence-string=EXCERPT OF UNWANTED MESSAGE

Filter out messages that contain the given string (you only need to provide part of the message text for the whole message to be suppressed). Use this option multiple times to suppress several different messages. The string should be a single line, but that is not a problem for multiline log messages: space characters in the provided string match any sequence of whitespace characters in the message, including newlines. If needed, you may precede the string with “`///`”, in which case you can use lua-style pattern matching (<https://www.lua.org/pil/20.2.html>). Note that the string is used verbatim: you need not (and should not) enclose it in quotes nor escape special characters such as “`\`”.

--silence-file=FILENAME OR FILE GLOB

Filter out messages that have been generated while the given file was being processed. Do **not** use absolute or relative paths, only filenames. Simple file globs, such as “`*.cls`”, work as expected. Use this option multiple times to suppress messages from several different files.

--semisilence-file=FILENAME OR FILE GLOB

Just like the previous option, but non-recursive. This means that messages generated while the given file was being processed are excluded, but messages generated by some other file that was opened by it are not. For example, if “`chapters.tex`” includes (with \input) the files “`chapter1.tex`” and “`chapter2.tex`”, using “`--silence-file=chapters.tex`” will prevent messages generated by any of the three files from being displayed. If, however, you use “`--semisilence-file=chapters.tex`”, messages generated by `chapters.tex` will be suppressed, but messages generated by `chapter1.tex` or `chapter2.tex` will not.

---add-[debug|info|warning]-message=MESSAGE

Add MESSAGE to the list of messages known to the program with the given severity level; see Section UNRECOGNIZED MESSAGES below for more information about this. Like `--silence-string`, these should be a single line; unlike `--silence-string`, you need to embed `\n` explicitly to indicate line breaks. You may precede the string with `“////”` to use lua-style pattern matching, but embedding `\n` to indicate line breaks is unavoidable. Use these options multiple times to add many different messages.

-c CFGFILE, --config-file=CFGFILE

Read options from the given configuration file. Options are exactly the same as the long command-line options, but without the preceding `“--”` characters. Lines starting with a `“#”` sign are comments. An example configuration file:

```
no-page-delay
# no-page-delay enables shipouts, but we do not want that
no-shipouts
silence-string = Hyperreferences in rotated content will be misplaced
# no need to escape the "\" (or any other) character
silence-string = Using \overbracket and \underbracket from `mathtool
# silence a string using lua pattern matching
silence-string = ///luaotfload | aux : font no %d+ %(.-%)
silence-files = *.cls
```

-h, --help

Show concise options description.

--version

Print program version.

UNRECOGNIZED MESSAGES

texlogsieve automatically handles messages such as `“Package blah Info:...”` or `“LaTeX Warning:...”`. However, many messages do not follow this pattern. To do its thing, texlogsieve should know about these other messages beforehand. This is important for three reasons:

1. Unknown messages are given maximum priority; if you do not want to see them, you have to use `--silence-string`;
2. If the message has more than one line, each line is treated as an independent message. This means you need to use `--silence-string` multiple times;
3. In some rare cases, the line unwrapping algorithm may fail near an unrecognized message.

While texlogsieve recognizes quite a few messages out of the box, you may run into a message generated by some package that it does not know about (you can check for this using `“-l unknown”`). If that is the case, you can use the `---add-[debug|info|warning]-message` options to add it to the list of messages known to the program.

LIMITATIONS

texlogsieve does not try to do anything smart about error messages (at least for now); if there is an error, you probably want to take a look directly at the log file anyway.

Since it needs to know what messages to expect, texlogsieve is currently geared towards LaTeX; I have no idea how it would work with ConTeXt or plain TeX. Still, adding support to them should not be too difficult.

SEE ALSO

The pdf documentation (in TeXLive, try *texdoc texlogsieve*).

If you want to know more about the TeX log file and the workings of the program, check the initial comments in the code.

BUGS AND DEVELOPMENT

<https://gitlab.com/lago/texlogsieve>

COPYRIGHT

Copyright © 2021 Nelson Lago <lago@ime.usp.br>

License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.