Texdoc

Find & view documentation in TeX Live

https://tug.org/texdoc/

Manuel Pégourié-Gonnard Takuto Asakura

v3.2.1 2020-02-20

1 Quick Guide

Texdoc is a command-line tool to find and view documents in TeX Live. Typing

```
texdoc ⟨name⟩
```

in your command line, you will see a document of the ⟨name⟩ package is popped up. Of course, you have to replace ⟨name⟩ with the actual name of a package. To look up the documentation of more than one packages at once, just give multiple ⟨name⟩s as arguments.

1.1 Modes

Texdoc has several modes that determine how results will be returned. The default is view mode, which opens the first, that is supposedly the best, result with a suitable viewer. It is rather handy when you know what you want to read. On the other hand, there may be other relevant documents for the given ⟨name⟩, which are ignored in the view mode.

In the list mode, Texdoc lists all relevant documentation and ask you which one you want to view. This mode is useful when there are other interesting sources of information in addition to the main documentation of a package.

There is also a mixed mode, intended to get the best of the view mode and list mode: if there is only one good result, then Texdoc opens it in a viewer, like in the view mode. Otherwise, it offers you a menu, like in the list mode.

By default, Texdoc hides some results, which expected to be less relevant, unless it cannot find any relevant result. In the showall mode, Texdoc always shows all results, including bad ones.

A couple of command-line options are available for selecting the mode to execute: `-w` (`--view`) for the view mode, `-m` (`--mixed`) for the mixed mode, `-l` (`--list`) for the list mode, and `-s` (`--showall`) for the showall mode.

If you have your favorite mode and always use it, you may not want to keep typing the same option. The next section describes how to customize Texdoc using its configurations files.

1.2 Configuration files

The configuration file enables you to tweak Texdoc in many ways. You can use the `--files` option to know where to put your personal configuration file; you may need to create this file, possibly with some parent directories. If you want to know the full list of possible configuration files, see Section 2.3.

To set your favorite mode, just insert a line `mode =` ⟨mode⟩ in your personal configuration file, where ⟨mode⟩ is one of `view`, `mixed`, `list`, and `showall`. Though your preferred language is usually detected automatically by getting the system locale, you can set it explicitly with a line `lang =` ⟨2-letter code⟩ in the configuration file.

1.3 Viewers

The way of Texdoc for choosing a viewer varies according to your platform. On Windows, macOS, or Unix with KDE, GNOME, or XFCE, it uses your file associations like when you double-click files in the Explorer, the Finder or your default file manager (except for the text viewer, which is always a pager). Otherwise, it tries to find a viewer in the path from a list of “known” viewers.

You may want to use a different viewer for some types of documents. This can be achieved by setting the various `viewer_`⟨ext⟩ configuration items, where ⟨ext⟩ is an extension corresponding to a file type. For example, if you want to set `xpdf` as your default PDF viewer, and run it in the background, insert the line `viewer_pdf = xpdf %s &` in your configuration file. Herein, `%s` is a place holder for the name of the file to view.

You can stop reading now

The following parts explain the mechanisms of Texdoc for finding the best results and how to customize them. We have been trying hard to optimize the default configuration values so that normal users do not need to fiddle with it. Thus, the following parts are useful only when you are curious or have special needs.

2 Controlling Texdoc

2.1 Command-line options

The full usage of Texdoc can be summarized as follows:

```
texdoc ⟨names⟩
texdoc ⟨options⟩ ⟨names⟩
texdoc ⟨action⟩
texdoc ⟨options⟩ ⟨action⟩
```

We have tried to implement a GNU-compatible option parser. Short options, each of which consists of a single letter, must start with a single hyphen `-`. Multiple short options can be specified with a single hyphen, e.g., `-vl` is equivalent to `-v -l`. Long options have to be following double hyphens `--`. All options must be specified before the first argument. A String beginning with a hyphen after the first argument will be treated as an argument starting with a hyphen.

Some options are called action options, which can be used for ⟨action⟩ in the above usage. Four actions are available:

`-h`, `--help`

Shows a quick help message (namely a list of command-line options) and exit successfully.

`-V`, `--version`

Shows the current version of the program and exit successfully.

`-f`, `--files`

Shows the list of configuration files for the current installation and platform, with their status and exit successfully. Normally, only “active” and “disabled” files are shown (see `lastfile_switch`). To show “not found” files as well, you can use `--verbose`.

`--just-view` ⟨file⟩

Open ⟨file⟩ in the usual viewer. The file should be given with full path, absolutely no searching is done. This option is not really meant for users, but rather intended to be used from another program, like a GUI front-end to Texdoc.

Some normal options such as `-v` are effective for some actions but note that you have to specify such options before the action option. Options after an action option will be ignored.

The followings are other normal command-line options:

`-w`, `--view` (default)

Set `mode` to `view`. Texdoc will open the best results it found by the best application it knows. See Section 1.1.

`-l`, `--list`

Set `mode` to `list`. Texdoc will show the lists of all relevant documents it found and ask you which to open. When used with the `--nointeract` option, Texdoc will just show the lists of documents and exist successfully.

`-m`, `--mixed`

Set `mode` to `mixed`. If Texdoc finds only a document for your keyword, then it opens it, and if it finds multiple, then it will behave like in the `list` mode.

`-s`, `--showall`

Set `mode` to `showall`. It also behave like in a `list` mode, but show all the documents it found including documents which seem to be not a good result.

`-i`, `--interact` (default)

Texdoc may require user reactions, e.g., ask you which document to open in the `list` mode. Internally, this sets `interact_switch` to `true`.

`-I`, `--nointeract`

If this is specified, Texdoc will never request any input from users. Internally, this sets `interact_switch` to `false`.

`-M`, `--machine`

This make the result list of Texdoc machine-readable. Only effective for the `list` mode and its friends. This option implies `--nointeract`. Internally, this sets `machine_switch` to `true`.

`-q`, `--quiet`

This suppress warnings and most error messages. Internally, this sets `verbosity_level` to minimum.

`-v`, `--verbose`

Make Texdoc to print additional information such as viewer commands which are invoked by the program. Internally, this sets `verbosity_level` to maximum.

`-d` ⟨list⟩, `--debug=`⟨list⟩, `-D`, `--debug`

This sets `debug_list` to show dubbging information in the specified category. You can specify multiple categories with a comma-separated list. If you specify `-D` or `--debug` without specifying a list, it activates all available debug categories.

`-c` ⟨name⟩=⟨value⟩

This enables you to set arbitrary configuration items ⟨name⟩ to ⟨value⟩ from the command-line. See Section 4 for the list of all available configuration items.

2.2 Environment variables

Some environment variables affect the behavior of Texdoc. The environment variables can be roughly categorized into three groups.

First, as well as other programs in TeX Live, searching path for the Texdoc are determined with the Kpathsea variables like `TEXMF*`. The mechanism of Kpathsea is complex, so please refer to 'the TeX Live Guide', which supposed to be open by `texdoc texlive`, and the document of Kpathsea for the details. Usually, Texdoc searches documents for the `TEXMF` trees including `TEXMFDIST`, `TEXMFLOCAL`, and `TEXMFHOME` and trees specified in `TEXDOCS`. It also self-locates its own program using `TEXMF` search path. In addition, Texdoc uses `TEXMFROOT` and `TEXMFVAR` to look for the tlpdb database and the cache respectively.

Second, the viewers for opening documents can be controlled by some environment variables. They all correspond to some viewer_⟨ext⟩ setting.[1] You can append _texdoc to every name in the first column: this wins over every other name. These variables can be split by colon : and the first non-nil occurrence is used. If a viewer command contains colon, please specify it by viewer_⟨ext⟩.

Variable	Use for	Configuration item
BROWSER	HTML files	viewer_html
DVIVIEWER	DVI files	viewer_dvi
MDVIEWER	Markdown files	viewer_md
PAGER	Text files	viewer_txt
PDFVIEWER	PDF files	viewer_pdf
PSVIEWER	PS files	viewer_ps

Third, on Unix systems, locale-related variables such as LANG and LC_ALL are used for the default value of lang.

2.3 Precedence of configuration sources

Values for a particular setting can come from several sources. The sources are looked at in the following order and the first value found is always used:

1. Command-line options.
2. Environment variables ending with _texdoc.
3. Other environment variables.
4. Values from configuration files (see below).
5. Hard-coded defaults that may depend on the current machine.

The configuration files are found in the directories TEXMF/texdoc, where TEXMF is the kpathsea variable, in the order given by this variable. Inside each directory, three files are recognized, in this order:

1. texdoc-⟨platform⟩.cnf where ⟨platform⟩ is the name of the current platform (defined as the name of the directories where the TeX Live binaries are located, for example x86-64-linux). This may be useful when an installation is shared across machines with different architectures needing different settings, for example for viewers. Their use is not recommended in any other situation.
2. texdoc.cnf is the recommended file for normal use.

[1] Old names of environment variables, namely TEXDOCVIEW_{html,dvi,md,txt,pdf,ps} and TEXDOC_VIEWER_{HTML,DVI,MD,TXT,PDF,PS}, are deprecated but still work.

3. `texdoc-dist.cnf` is useful for installing a newer version of texdoc (including its default configuration file) in your home while retaining the use of the previous file for your personal setting; see GitHub repository for instructions on running the development version.

2.4 Exit codes

The status of Texdoc, which is whether or not it was successfully executed and an expected result was delivered, can be grasped by seeing its exit code. This will be useful to manage Texdoc by other programs. The current exit codes of Texdoc are:

`0` Success.
`1` Internal error.
`2` Usage error.
`3` No documentation found.

3 Customizing the Search Results

3.1 An overview of how Texdoc works

When you type `texdoc` ⟨keyword⟩, first filenames related to the ⟨keyword⟩ are collected from the two following sources.

1. the `TEXMF` trees containing documentation that specified by the kpathsea variable `TEXDOCS`: Texdoc collects all files containing ⟨keyword⟩ in their filenames or in the names of their parent directories.
2. the TeX Live Database (`texlive.tlpdb`): Texdoc searches for packages named ⟨keyword⟩ or containing a file ⟨keyword⟩`.`⟨ext⟩ where ⟨ext⟩ may be `sty` or `cls`, and collects all document files of the found packages.

Second, the collected filenames are filtered by their extensions. After filtering, only files with known extensions (listed in `ext_list`) remain. If Texdoc cannot find any documentation, the fuzzy search will find the closest package name to the ⟨keyword⟩ and it will recollect document files (see Section 3.6).

Third, the collected filenames are given numeric scores by using some heuristics. For example, a filename ⟨keyword⟩`.pdf` is good, and thus gets a high score. A name ⟨keyword⟩`-`⟨lang⟩`.pdf` is also good and gets a higher score if the ⟨lang⟩ is the preferred language code in your configuration. In addition, there are several rules such as “a name ⟨keyword⟩`-doc` always wins over any names ⟨keyword⟩`whatever`” and “files under a directory named exactly ⟨keyword⟩ get bonuses”. Scores may also be adjusted based on the

extensions and some known names or subwords. For example, by default, `Makefile`s get very low scores because basically they are not documents.[2]

Finally, a file with the highest score is opened with a viewer, or the list of results is shown, depending on the current mode. Usually, only results with positive scores are displayed, except in the `showall` mode. Results with very bad scores, that are lower than `-100`, are never displayed.

This model for searching and scoring is efficient, but unfortunately, it is not perfect: Texdoc may sometimes need a hint, either to find a relevant file or, more likely, to recognize which of the found files is the best. For example, suppose you are looking for a document of the LaTeX package `shortvrb`. Texdoc will successfully find `shortvrb.sty` in the TeX Live package `latex`. However, Texdoc will not be able to sort a number of documents in the `latex` package correctly because none of them contains the string `shortvrb`.

For such cases, aliases are useful: in the default configuration file, `shortvrb` is aliased to `base/doc`, so that Texdoc primarily look for `base/doc` when you type `texdoc shortvrb`. Note that Texdoc will also look for the original keyword, and that a name can be aliased to more than one new name.

3.2 Aliases

Aliases can be set in configuration files of Texdoc with the following syntax:

```
alias ⟨original keyword⟩ = ⟨name⟩
alias(⟨score⟩) ⟨original keyword⟩ = ⟨name⟩
```

A number of aliases are already set in the default configuration file. You can also define your own aliases in your personal configuration files (see Section 1.2 or 2.3 to know where is that). For example, you can add a line

```
alias td = texdoc
```

to the file to alias `td` to `texdoc`. Then, files in the doc trees matching exactly with `texdoc`, i.e., this document `texdoc.pdf`, will be added to the result list when you search for the keyword `td`.

By default, files match with the aliased name get a score of `10`. This score is greater than any result of heuristic scoring. It means that results found via aliases will always be ranked higher than results found for the original keyword. If you want the results associated with a particular alias to have

[2] Nevertheless, they often end up in the `doc` trees, because in TeX Live, sources of documents are usually in the same directory as the documents themselves. Other source files for documents are discriminated by their extensions.

a custom score instead of the default score `10`, you can use the optional argument to the `alias` directive. This can be useful if you associate multiple aliases to a keyword and want to specify the priorities for them.

In addition, aliases for ⟨keyword⟩`-`⟨lang⟩, where ⟨lang⟩ is your preferred language's 2-letter code (as detected or configured, see the `lang` option), are also used when searching for ⟨keyword⟩. Files match with these language-based aliases get additional bonus scores so that they win over the results for language-independent aliases.

For concrete examples, please have a look at the default configuration file, which can be found in the last line of the output for `texdoc -f`. If you think aliases you defined locally should be added to the default configuration, please share the idea to the mailing list or GitHub repository.

Aliases are additive: if you define your own aliases for a keyword in your configuration file, and there are also aliases for the same keyword in the default configuration, they will add up. To prevent the default aliases from being applied for a particular keyword, add a line `stopalias` ⟨keyword⟩ in your personal configuration file. It will preserve the aliases defined before this directive if any, but prevent all further aliasing on this keyword.

Here are some notes for aliases. First, they are case-insensitive. Second, they do not work recursively: only aliases set to the original keyword are used. Third, results found for aliases always get the score defined by the `alias` directive (`10` by default). The adjustments described in the other subsections are not applied to results for aliases.

3.3 Score adjustments

Scores can be arbitrary adjusted by using the `adjscore` directive in the configuration files. The adjustment can be done either for global, i.e., for any keywords, or a particular keyword. The syntax is as follow:

```
adjscore ⟨pattern⟩ = ⟨score adjustment⟩
adjscore(⟨keyword⟩) ⟨pattern⟩ = ⟨score adjustment⟩
```

When the optional argument ⟨keyword⟩ is specified, the adjustment will be applied only when a user searches for the ⟨keyword⟩. Otherwise, the adjustment applied always for the ⟨pattern⟩.

Here are a few examples from the default configuration file.

```
adjscore /Makefile = -1000
adjscore /tex-virtual-academy-pl/ = -50
adjscore(tex) texdoc = -10
```

Herein, all files named `Makefile`, and those with suffixes, e.g., `Makefile.in`, will be eliminated from results: by adjusting their score with such a large negative value, their final score will be less than `-100`. Which means they will never be displayed. Files from the `tex-virtual-academy-pl` directory, on the other hand, are not eliminated but get a negative adjustment, since they are a common source of "fake" matches which hide better results. The third line gives a minus adjustment for results containing `texdoc` only when the searched keyword is `tex`. Otherwise, such results would get high scores because the heuristic scoring would assume `texdoc` is the name of documentation for TEX itself. The adjustment value `-10` is enough to ensure that those results will have a negative score, so the files containing `texdoc` will not be displayed unless `showall` mode is used.

Note that the values of scores, such as the default score for aliases and the range of heuristic scores, may be changed in a future version of Texdoc. Therefore, you may need to adapt your score adjustments after updating Texdoc in the future.

3.4 Extensions and basenames of files

File extensions regarded as documents by Texdoc can be specified with the configuration item `ext_list`. By default, files with extensions `pdf`, `html`, `htm`, `txt`, `md`, `ps`, and `dvi` and files without extension are recognized as documents.

During the scoring process, the configuration item `badext_list` is also used: files with a "bad" extension appearing in this list will get a lesser score.

In the practical filenames, things which are not actually extensions can follow dots, e.g., `readme.fr` and `readme.texlive`. Thus, in addition to the lists of known extensions, Texdoc has lists for known basenames: `basename_list` and `badbasename_list`. These configuration items are respectively similar to `ext_list` and `badext_list`. In short, a file will be selected if either its extension or its basename is known, and get a lesser score if either is known to be "bad".

3.5 Common filenames

Document filenames of packages are often in the form of ⟨package⟩`-doc`. Respecting this culture in the TEX community, Texdoc gives a special score for files named ⟨package⟩⟨suffix⟩, where ⟨suffix⟩ is an element of the list `suffix_list`. Please refer to the shipped configuration file for the default setting. Also, feel free to suggest additional elements for the list to the mailing list with a real-life example.

3.6 Fuzzy search

When the normal search cannot find any document in TeX Live, Texdoc will execute a fuzzy search. The fuzzy search tries to find the closest name of a package in TeX Live[3] to the input ⟨keyword⟩. The results of the fuzzy search are shown in an info message, which can be seen by using the command-line option `-v`.

The default allowance of Levenshtein distance is 5. You can change this allowance by using the configuration item `fuzzy_level`. Results of fuzzy search could be different among executions if multiple package names have the same Levenshtein distance to the input.

4 Configuration items

Configuration files are line-oriented text files. Comments begin with a # and run to the end of line. Lines containing only space are ignored. Space at the beginning or end of a line, as well as around an = sign, is ignored. Apart from comments and empty lines, each line must be of one of the following forms.

```
⟨configuration item⟩ = ⟨value⟩
alias ⟨original keyword⟩ = ⟨name⟩
alias(⟨score⟩) ⟨original keyword⟩ = ⟨name⟩
stopalias ⟨original keyword⟩
adjscore ⟨pattern⟩ = ⟨score adjustment⟩
adjscore(⟨keyword⟩) ⟨pattern⟩ = ⟨score adjustment⟩
```

We will concentrate on the ⟨configuration item⟩ part here, since other directives have already been presented (Section 3.2 and 3.3).

In the above, ⟨value⟩ never needs to be quoted: quotes would be interpreted as part of the value, not as quotation marks (this also holds for the other directives).

Lines which do not obey these rules raise a warning, as well as unrecognised values of ⟨configuration item⟩. The ⟨value⟩ can be an arbitrary string, except when the name of the ⟨configuration item⟩ ends with:

1. `_list`, then ⟨value⟩ is a coma-separated list of strings. Spaces around commas are ignored. Two consecutive comas or a coma at the beginning or end of the list means the empty string at the corresponding place.
2. `_switch`, then ⟨value⟩ must be either `true` or `false` (lowercase).
3. `_level` and `_lines`, then ⟨value⟩ is an integer.

[3] Note that the feature searches only package names at this point. Other objects such as aliases cannot be found by the fuzzy search.

In these cases, an improper ⟨value⟩ will raise a warning too.

`mode = view | list | mixed | showall` (default: `view`)

Set the mode to the given value. The various modes have been described in Section 1.1.

`interact_switch = true | false` (default: `true`)

Turn on or off interaction. Turning interaction off prevents Texdoc from asking you to choose a file to view when there are multiple choices, so it just prints the list of files found.

`suffix_list =` ⟨list⟩ (default: empty)

Set the list of known suffixes to ⟨list⟩ (see Section 3.5). Default is the empty list, but see the shipped configuration file for more.

`ext_list =` ⟨list⟩ (default: `pdf, html, htm, txt, md, dvi, ps,`)

Set the list of recognised extensions to ⟨list⟩. This list is used to filter and sort the results that have the same score (with the default value: pdf first, etc). Two special values are recognised:

- The empty element. This means files without extensions, or more precisely without a dot in their name. This is meant for files like `README`, etc. The file is assumed to be plain text for viewing purpose.
- `*` means any extension. Of course if it is present in the list, it can be the only element!

There is a very special case: if the searched ⟨name⟩ has `.sty` extension, Texdoc enters a special search mode for `.sty` files (not located in the same place as real documentation files) for this ⟨name⟩, independently of the current value of `ext_list` and `mode`. In an ideal world, this wouldn't be necessary since every sty file would have a proper documentation in pdf, html or plain text, but…
For each ⟨ext⟩ in `ext_list` there should be a corresponding `viewer_`⟨ext⟩ value set. Defaults are defined corresponding to the default `ext_list`, but you can add values if you want. For example, if you want Texdoc to be able to find man pages and display them with the `man` command, you can use

```
ext_list = pdf, html, htm, 1, 5, txt, md, dvi, ps,
viewer_1 = man
viewer_5 = man
```

As a special case, if the extension is `sty`, then the `txt` viewer is used; similarly, if it is `htm` the `html` viewer is used. Otherwise, the `txt` viewer is used and a warning is issued.

`badext_list = ⟨list⟩` (default: `txt`)

Set the list of "bad" extensions to ⟨list⟩. Files with those extensions get a malus of `1` on their heuristic score if it was previously positive.

`basename_list = ⟨list⟩` (default: `readme, 00readme`)

Set the list of "known" base names to ⟨list⟩. Files with those base names are selected regardless of their extension. If the extension is unknown, the text viewer will be used to view the file.

`badbasename_list = ⟨list⟩` (default: `readme, 00readme`)

Set the list of "bad" base names to ⟨list⟩. Files with those names get a malus of `1` on their heuristic score if it was previously positive.

`viewer_⟨ext⟩ = ⟨command⟩`

Set the viewer command for files with extension ⟨ext⟩ to ⟨command⟩. For files without an extension, `viewer_txt` is used. Note that there is no `viewer_` variable. In ⟨command⟩, `%s` can be used as a placeholder for the filename, which is otherwise inserted at the end of the command. The command can be an arbitrary shell construct.

`lang = ⟨2-letter code⟩`

Set your preferred language. By default, it reflects your system's locale.

`verbosity_level = ⟨number⟩` (default: `2`)

Set the verbosity level to ⟨number⟩. At level `3`, errors, warnings and informational messages will be printed on stderr; `2` means only errors and warnings, `1` only errors and `0` nothing except internal errors (obviously not recommended).

`debug_list` = ⟨list⟩ (default: empty)

Set the list of activated debug categories. Available debug categories are `config`, `files`, `search`, `score`, `texdocs`, `tlpdb`, `version`, `view`, and `all` to activate all of these. Implies `--verbose`. Debug information are printed on standard error.

`max_lines` = ⟨number⟩ (default: `20`)

Set the maximum number of results to be printed without confirmation in list, mixed or showall mode. This setting has no effect if interaction is disabled.

`machine_switch` = true | false (default: `false`)

Turn on or off machine-readable output. With this option active, the value of `interact_switch` is forced to `false`, and each line of output is

⟨argument⟩`\t`⟨score⟩`\t`⟨filename⟩

where ⟨argument⟩ is the name of the argument to which the results correspond (mainly useful if there were many arguments), `\t` is the tab (ASCII #9) character, and the other entries are pretty self-explanatory. Nothing else is printed on stdout, except if an internal error occurs (in which case exit code will be 1). In the future, more tab-separated fields may be added at the end of the line, but the first 3 fields will remain unchanged. Currently, there are two additional fields: a two-letter language code, and an unstructured description, both taken from the CTAN catalogue (via the TeX Live database). These fields may be empty, and they are not guaranteed to keep the same meaning in future versions of Texdoc.

`zipext_list` = ⟨list⟩ (default: empty)

List of supported extensions for zipped files. Allows compressed files with names like `foobar.`⟨zip⟩, with ⟨zip⟩ in the given ⟨list⟩, to be found and unzipped before the viewer is started (the temporary file will be destroyed right after).
Warning: Support for zipped documentation is not meant to work on windows, a Unix shell is assumed! If you add anything to this list, please make sure that you also set a corresponding `unzip=`⟨ext⟩ value for each ⟨ext⟩ in the list. At the same time, make sure you are using blocking (i.e., not returning immediately) viewers.
Remark: TeX Live doesn't ship compressed documentation files, so this option is mainly useful with re-packaged version of TeX Live that do, for example in Linux distributions.

`unzip_⟨zipext⟩ = ⟨command⟩` (no default)

The unzipping command for compressed files with extension ⟨zipext⟩. Define one for each item in `zipext_list`. The command must print the result on stdout, like `gzip -d -c` does.

`rm_file = ⟨command⟩` (default: `rm -f`)

Commands for removing files on your system. Only useful for zipped documents (see `zipext_list`).

`rm_dir = ⟨command⟩` (default: `rmdir`)

Commands for removing directories on your system. Also only useful for zipped documents (see `zipext_list`).

`lastfile_switch = true | false` (default: `false`)

If set to true, prevents Texdoc from reading any other configuration file after this one (they will be reported as "disabled" by `texdoc -f`). Mainly useful for installing a newer version of Texdoc in your home and preventing the default configuration file from older versions to be used (see the README for instructions on how to do so).

`fuzzy_level = ⟨number⟩` (default: `5`)

Set the allowance of Levenshtein distance to ⟨number⟩ for fuzzy search. At level `0`, the fuzzy search feature is disabled.

`texlive_tlpdb = ⟨path⟩` (no default)

This enables you to specify the ⟨path⟩ for the database `texlive.tlpdb` which Texdoc uses for searching documents. By default, Texdoc uses `TEXMFROOT/tlpkg/texlive.tlpdb`. Usually, you don't have to (or should not) set this configuration item. This is only useful when you want to use Texdoc within a TeX Live-based TeX distribution which does not ship `texlive.tlpdb`.

5 Licence

Previous work (Texdoc program) in the public domain:

- Contributions from Reinhard Kotucha (2008).
- First texlua versions by Frank Küster (2007).
- Original shell script version by Thomas Esser, David Aspinall, and Simon Wilkinson.

Happy TEXing!