Package ‘tinytex’

December 13, 2022

Type Package

Title Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Documents

Version 0.43

Description Helper functions to install and maintain the 'LaTeX' distribution named 'TinyTeX' (<https://yihui.org/tinytex/>), a lightweight, cross-platform, portable, and easy-to-maintain version of 'TeX Live'. This package also contains helper functions to compile 'LaTeX' documents, and install missing 'LaTeX' packages automatically.

Imports xfun (>= 0.29)

Suggests testit, rstudioapi

License MIT + file LICENSE

URL https://github.com/rstudio/tinytex

BugReports https://github.com/rstudio/tinytex/issues

Encoding UTF-8

RoxygenNote 7.2.2

NeedsCompilation no

Author Yihui Xie [aut, cre, cph] (<https://orcid.org/0000-0003-0645-5666>), RStudio, PBC [cph], Christophe Dervieux [ctb] (<https://orcid.org/0000-0003-4474-2498>), Devon Ryan [ctb] (<https://orcid.org/0000-0002-8549-0971>), Ethan Heinzen [ctb], Fernando Cagua [ctb]

Maintainer Yihui Xie <xie@yihui.name>

Repository CRAN

Date/Publication 2022-12-13 07:10:02 UTC
check_installed

**Description**

If a package has been installed in TinyTeX or TeX Live, the command `tlmgr info PKG` should return `PKG` where `PKG` is the package name.

**Usage**

`check_installed(pkgs)`

**Arguments**

- `pkgs` A character vector of LaTeX package names.

**Value**

A logical vector indicating if packages specified in `pkgs` are installed.

**Note**

This function only works with LaTeX distributions based on TeX Live, such as TinyTeX.

**Examples**

```r
tinytex::check_installed('framed')
```

---

**check_installed**

Check if certain LaTeX packages are installed

**Description**

If a package has been installed in TinyTeX or TeX Live, the command `tlmgr info PKG` should return `PKG` where `PKG` is the package name.

**Usage**

`check_installed(pkgs)`

**Arguments**

- `pkgs` A character vector of LaTeX package names.

**Value**

A logical vector indicating if packages specified in `pkgs` are installed.

**Note**

This function only works with LaTeX distributions based on TeX Live, such as TinyTeX.

**Examples**

```r
tinytex::check_installed('framed')
```
**copy_tinytex**  
*Copy TinyTeX to another location and use it in another system*

**Description**

The function `copy_tinytex()` copies the existing TinyTeX installation to another directory (e.g., a portable device like a USB stick). The function `use_tinytex()` runs `tlmgr path add` to add the copy of TinyTeX in an existing folder to the `PATH` variable of the current system, so that you can use utilities such as `tlmgr` and `pdflatex`, etc.

**Usage**

```r
copy_tinytex(
  from = tinytex_root(),
  to = select_dir("Select Destination Directory")
)

use_tinytex(from = select_dir("Select TinyTeX Directory"))
```

**Arguments**

- `from`: The root directory of the TinyTeX installation. For `copy_tinytex()`, the default value `tinytex_root()` should be a reasonable guess if you installed TinyTeX via `install_tinytex()`. For `use_tinytex()`, if `from` is not provided, a dialog for choosing the directory interactively will pop up.

- `to`: The destination directory where you want to make a copy of TinyTeX. Like `from` in `use_tinytex()`, a dialog will pop up if `to` is not provided in `copy_tinytex()`.

**Note**

You can only copy TinyTeX and use it in the same system, e.g., the Windows version of TinyTeX only works on Windows.

---

**install_tinytex**  
*Install/Uninstall TinyTeX*

**Description**

The function `install_tinytex()` downloads and installs TinyTeX, a custom LaTeX distribution based on TeX Live. The function `uninstall_tinytex()` removes TinyTeX; `reinstall_tinytex()` reinstalls TinyTeX as well as previously installed LaTeX packages by default; `tinytex_root()` returns the root directory of TinyTeX if found.
install_tinytex

Usage

install_tinytex(
  force = FALSE,
  dir = "auto",
  version = "daily",
  bundle = "TinyTeX-1",
  repository = "auto",
  extra_packages = if (is_tinytex()) tl_pkgs(),
  add_path = TRUE
)

uninstall_tinytex(force = FALSE, dir = tinytex_root())

reinstall_tinytex(packages = TRUE, dir = tinytex_root(), ...)

tinytex_root(error = TRUE)

Arguments

force Whether to force to install or uninstall TinyTeX. For install_tinytex(), force = FALSE will stop this function from installing TinyTeX if another LaTeX distribution is detected, or the directory specified via the dir argument exists.
dir The directory to install (should not exist unless force = TRUE) or uninstall TinyTeX.
version The version of TinyTeX, e.g., "2020.09" (see all available versions at https://github.com/rstudio/tinytex-releases, or via xfun::github_releases("rstudio/tinytex-releases"). By default, it installs the latest daily build of TinyTeX. If version = 'latest', it installs the latest monthly Github release of TinyTeX.
bundle The bundle name of TinyTeX (which determines the collection of LaTeX packages to install). See https://github.com/rstudio/tinytex-releases#releases for all possible bundles and their meanings.
repository The CTAN repository to set. By default, it is the repository automatically chosen by https://mirror.ctan.org (which is usually the fastest one to your location). You can find available repositories at https://ctan.org/mirrors, e.g., 'http://mirrors.tuna.tsinghua.edu.cn/CTAN/', or 'https://mirror.las.iastate.edu/tex-archive/'. In theory, this argument should end with the path '/systems/texlive/tlnet', and if it does not, the path will be automatically appended.
extra_packages A character vector of extra LaTeX packages to be installed. By default, a vector of all currently installed LaTeX packages if an existing installation of TinyTeX is found. If you want a fresh installation, you may use extra_packages = NULL.
add_path Whether to run the command tlmgr path add to add the bin path of TeX Live to the system environment variable PATH.
packages Whether to reinstall all currently installed packages.
... Other arguments to be passed to install_tinytex() (note that the extra_packages argument will be set to tl_pkgs() if packages = TRUE).
error Whether to signal an error if TinyTeX is not found.
Note

If you really want to disable the installation, you may set the environment variable `TINYTEX_PREVENT_INSTALL` to `true`. Then `install_tinytex()` will fail immediately. This can be useful to sysadmins who want to prevent the accidental installation of TinyTeX.

References

See the TinyTeX documentation (https://yihui.org/tinytex/) for the default installation directories on different platforms.

---

**is_tinytex**

Check if the LaTeX installation is TinyTeX

---

**Description**

First find the root directory of the installation via `tinytex_root()`. Then check if the directory name is "tinytex" (case-insensitive). If not, further check if the first line of the file `texmf-dist/web2c/fmtutil.cnf` under the directory contains "TinyTeX" or ".TinyTeX". If the binary version of TinyTeX was installed, `fmtutil.cnf` should contain a line like ‘Generated by */TinyTeX/bin/x86_64-darwin/tlmgr on Thu Sep 17 07:13:28 2020’.

**Usage**

`is_tinytex()`

**Value**

A logical value indicating if the LaTeX installation is TinyTeX.

**Examples**

`tinytex::is_tinytex()`

---

**latexmk**

Compile a LaTeX document

---

**Description**

The function `latexmk()` emulates the system command `latexmk` (https://ctan.org/pkg/latexmk) to compile a LaTeX document. The functions `pdflatex()`, `xelatex()`, and `lualatex()` are wrappers of `latexmk(engine = , emulation = TRUE)`.
Usage

latexmk(
  file,
  engine = c("pdflatex", "xelatex", "lualatex", "latex", "tectonic"),
  bib_engine = c("bibtex", "biber"),
  engine_args = NULL,
  emulation = TRUE,
  min_times = 1,
  max_times = 10,
  install_packages = emulation && tlmgr_writable(),
  pdf_file = gsub("tex$", "pdf", file),
  clean = TRUE
)

pdflatex(...)
xelatex(...)
lualatex(...)

Arguments

file A LaTeX file path.

engine A LaTeX engine (can be set in the global option tinytex.engine, e.g., options(tinytex.engine = 'xelatex')).

bib_engine A bibliography engine (can be set in the global option tinytex.bib_engine).

ingenGINE_ARGS Command-line arguments to be passed to engine (can be set in the global option tinytex.engine_args, e.g., options(tinytex.engine_args = '-shell-escape').

emulation Whether to emulate the executable latexmk using R. Note that this is unused when engine == 'tectonic'.

min_times, max_times The minimum and maximum number of times to rerun the LaTeX engine when using emulation. You can set the global options tinytex.compile.min_times or tinytex.compile.max_times, e.g., options(tinytex.compile.max_times = 3).

install_packages Whether to automatically install missing LaTeX packages found by parse_packages() from the LaTeX log. This argument is only for the emulation mode and TeX Live. Its value can also be set via the global option tinytex.install_packages, e.g., options(tinytex.install_packages = FALSE).

pdf_file Path to the PDF output file. By default, it is under the same directory as the input file and also has the same base name. When engine == 'latex', this will be a DVI file.

clean Whether to clean up auxiliary files after compilation (can be set in the global option tinytex.clean, which defaults to TRUE).

... Arguments to be passed to latexmk() (other than engine and emulation).
The \texttt{latexmk} emulation works like this: run the \LaTeX{} engine once (e.g., \texttt{pdflatex}), run \texttt{makeindex} to make the index if necessary (the \texttt{*.idx} file exists), run the bibliography engine \texttt{bibtex} or \texttt{biber} to make the bibliography if necessary (the \texttt{*.aux} or \texttt{*.bcf} file exists), and finally run the \LaTeX{} engine a number of times (the maximum is 10 by default) to resolve all cross-references.

By default, \LaTeX{} warnings will be converted to R warnings. To suppress these warnings, set \texttt{options(tinytex.latexmk.warning = FALSE)}.

If \texttt{emulation = FALSE}, you need to make sure the executable \texttt{latexmk} is available in your system, otherwise \texttt{latexmk()} will fall back to \texttt{emulation = TRUE}. You can set the global option \texttt{options(tinytex.latexmk.emulation = FALSE)} to always avoid emulation (i.e., always use the executable \texttt{latexmk}).

The default command to generate the index (if necessary) is \texttt{makeindex}. To change it to a different command (e.g., \texttt{zhmakeindex}), you may set the global option \texttt{tinytex.makeindex}. To pass additional command-line arguments to the command, you may set the global option \texttt{tinytex.makeindex.args} (e.g., \texttt{options(tinytex.makeindex = 'zhmakeindex', tinytex.makeindex.args = c('-z', 'pinyin'))}).

If you are using the \LaTeX{} distribution Tiny\TeX, but its path is not in the \texttt{PATH} variable of your operating system, you may set the global option \texttt{tinytex.tlmgr.path} to the full path of the executable \texttt{tlmgr}, so that \texttt{latexmk()} knows where to find executables like \texttt{pdflatex}. For example, if you are using Windows and your Tiny\TeX{} is on an external drive ‘Z:/’ under the folder ‘Tiny\TeX’, you may set \texttt{options(tinytex.tlmgr.path = 'Z:/Tiny\TeX/bin/win32/tlmgr.bat')}. Usually you should not need to set this option because Tiny\TeX{} can add itself to the \texttt{PATH} variable during installation or via \texttt{use_tinytex()}. In case both methods fail, you can use this manual approach.

**Value**

A character string of the path of the output file (i.e., the value of the \texttt{pdf_file} argument).

---

**parse_install**

*Parse the \LaTeX{} log and install missing \LaTeX{} packages if possible*

**Description**

This is a helper function that combines \texttt{parse_packages()} and \texttt{tlmgr_install()}.

**Usage**

\texttt{parse\_install(...)}

**Arguments**

\texttt{...}  Arguments passed to \texttt{parse_packages()}.  


parse_packages  Find missing LaTeX packages from a LaTeX log file

Description

Analyze the error messages in a LaTeX log file to figure out the names of missing LaTeX packages that caused the errors. These packages can be installed via `tlmgr_install()`. Searching for missing packages is based on `tlmgr_search()`.

Usage

```r
parse_packages(
  log,
  text = readLines(log),
  files = detect_files(text),
  quiet = rep(FALSE, 3)
)
```

Arguments

- `log`  
  Path to the LaTeX log file (typically named ‘*.log’).
- `text`  
  A character vector of the error log (read from the file provided by the `log` argument by default).
- `files`  
  A character vector of names of the missing files (automatically detected from the `log` by default).
- `quiet`  
  Whether to suppress messages when finding packages. It should be a logical vector of length 3: the first element indicates whether to suppress the message when no missing LaTeX packages could be detected from the log, the second element indicate whether to suppress the message when searching for packages via `tlmgr_search()`, and the third element indicates whether to warn if no packages could be found via `tlmgr_search()`.

Value

A character vector of LaTeX package names.

r_texmf  Add/remove R’s texmf tree to/from TeX Live

Description

R ships a custom texmf tree containing a few LaTeX style and class files, which are required when compiling R packages manuals (‘Rd.sty’) or Sweave documents (‘Sweave.sty’). This tree can be found under the directory `file.path(R.home('share'), 'texmf')`. This function can be used to add/remove R’s texmf tree to/from TeX Live via `tlmgr_conf('auxtrees')`.
Usage

r_texmf(action = c("add", "remove"), ...)

Arguments

action  Add/remove R’s texmf tree to/from TeX Live.
...
Arguments passed to tlmgr().

References

See the tlmgr manual for detailed information about tlmgr conf auxtrees. Check out https://tex.stackexchange.com/q/77720/9128 if you don’t know what texmf means.

Examples

# running the code below will modify your texmf tree; please do not run
# unless you know what it means

# r_texmf('remove')
# r_texmf('add')

# all files under R's texmf tree
list.files(file.path(R.home('share'), 'texmf'), recursive = TRUE, full.names = TRUE)

_tlmgr

Run the TeX Live Manager

Description

Execute the tlmgr command to search for LaTeX packages, install packages, update packages, and so on.

Usage

tlmgr(args = character(), usermode = FALSE, ..., .quiet = FALSE)
tlmgr_search(what, file = TRUE, all = FALSE, global = TRUE, word = FALSE, ...)
tlmgr_install(  
  pkgs = character(),
  usermode = FALSE,
  path = !usermode && os != "windows",
  ...
)
tlmgr_remove(pkgs = character(), usermode = FALSE)
tlmgr_version(raw = TRUE)

tlmgr_update(
  all = TRUE,
  self = TRUE,
  more_args = character(),
  usermode = FALSE,
  run_fmtutil = TRUE,
  delete_tlpdb = getOption("tinytex.delete_tlpdb", FALSE),
  ...
)

tlmgr_path(action = c("add", "remove"))

tlmgr_conf(more_args = character(), ...)

tlmgr_repo(url = NULL, ...)

Arguments

args  A character vector of arguments to be passed to the command tlmgr.

usermode  (For expert users only) Whether to use TeX Live’s user mode. If TRUE, you must have run tlmgr(‘init-usertree’) once before. This option allows you to manage a user-level texmf tree, e.g., install a LaTeX package to your home directory instead of the system directory, to which you do not have write permission. This option should not be needed on personal computers, and has some limitations, so please read the tlmgr manual very carefully before using it.

...  For tlmgr(), additional arguments to be passed to system2() (e.g., stdout = TRUE to capture stdout). For other functions, arguments to be passed to tlmgr().

.quiet  Whether to hide the actual command before executing it.

what  A search keyword as a (Perl) regular expression.

file  Whether to treat what as a filename (pattern).

all  For tlmgr_search(), whether to search in everything, including package names, descriptions, and filenames. For tlmgr_update(), whether to update all installed packages.

global  Whether to search the online TeX Live Database or locally.

word  Whether to restrict the search of package names and descriptions to match only full words.

pkgs  A character vector of LaTeX package names.

path  Whether to run tlmgr_path(‘add’) after installing packages (path = TRUE is a conservative default: it is only necessary to do this after a binary package is installed, such as the metafont package, which contains the executable mf, but it does not hurt even if no binary packages were installed).

.raw  Whether to return the raw output of the command tlmgr --version, or a short version of the format ‘TeX Live YEAR (TinyTeX) with tlmgr DATE’.
self  Whether to update the TeX Live Manager itself.
more_args  A character vector of more arguments to be passed to the command tlmgr update or tlmgr conf.
run_fmtutil  Whether to run fmtutil-sys --all to (re)create format and hyphenation files after updating tlmgr.
delete_tlpdb  Whether to delete the `texlive.tlpdb.HASH' files (where HASH is an MD5 hash) under the `tlpkg' directory of the root directory of TeX Live after updating.
action  On Unix, add/remove symlinks of binaries to/from the system’s PATH. On Windows, add/remove the path to the TeXLive binary directory to/from the system environment variable PATH.
url  The URL of the CTAN mirror. If NULL, show the current repository, otherwise set the repository. See the repository argument of install_tinytex() for examples.

Details

The tlmgr() function is a wrapper of system2('tlmgr'). All other tlmgr_*() functions are based on tlmgr for specific tasks. For example, tlmgr_install() runs the command tlmgr install to install LaTeX packages, and tlmgr_update runs the command tlmgr update, etc. Note that tlmgr_repos runs tlmgr options repository to query or set the CTAN repository. Please consult the tlmgr manual for full details.

References

The tlmgr manual: https://www.tug.org/texlive/doc/tlmgr.html

Examples

# search for a package that contains titling.sty
tlmgr_search('titling.sty')

# to match titling.sty exactly, add a slash before the keyword, e.g.
tlmgr_search('/titling.sty')

# use a regular expression if you want to be more precise, e.g.
tlmgr_search('/titling\.sty$')

# list all installed LaTeX packages
tlmgr(c('info', '--list', '--only-installed', '--data', 'name'))
**tl_pkgs**  
List the names of installed TeX Live packages

**Description**
Calls `tlmgr info --list --data name` to obtain the names of all (installed) TeX Live packages. Platform-specific strings in package names are removed, e.g., "tex" is returned for the package `tex.x86_64-darwin`.

**Usage**
```
tl_pkgs(only_installed = TRUE)
```

**Arguments**
- `only_installed`  Whether to list installed packages only.

**Value**
A character vector of package names.

---

**tl_sizes**  
Sizes of LaTeX packages in TeX Live

**Description**
Use the command `tlmgr info --list` to obtain the sizes of LaTeX packages.

**Usage**
```
tl_sizes(show_total = TRUE, pkgs = NULL, only_installed = TRUE, field = "size")
```

**Arguments**
- `show_total`  Whether to show the total size.
- `pkgs`  A character vector of package names (by default, all packages).
- `only_installed`  Whether to list installed packages only.
- `field`  A character vector of field names in the package information. See [https://www.tug.org/texlive/doc/tlmgr.html#info](https://www.tug.org/texlive/doc/tlmgr.html#info) for more info.

**Value**
By default, a data frame of three columns: `package` is the package names, `size` is the sizes in bytes, and `size_h` is the human-readable version of sizes. If different field names are provided in the `field` argument, the returned data frame will contain these columns.
Index

check_installed, 2
copy_tinytex, 3

install_tinytex, 3, 11
is_tinytex, 5

latexmk, 5
lualatex (latexmk), 5

parse_install, 7
parse_packages, 6, 7, 8
pdflatex (latexmk), 5

r_texmf, 8
reinstall_tinytex (install_tinytex), 3

system2, 10

tinytex_root, 5
tinytex_root (install_tinytex), 3
tl_pkgs, 12
tl_sizes, 12
tlmgr, 9, 9
tlmgr_conf, 8
tlmgr_conf (tlmgr), 9
tlmgr_install, 7, 8
tlmgr_install (tlmgr), 9
tlmgr_path (tlmgr), 9
tlmgr_remove (tlmgr), 9
tlmgr_repo (tlmgr), 9
tlmgr_search, 8
tlmgr_search (tlmgr), 9
tlmgr_update (tlmgr), 9
tlmgr_version (tlmgr), 9

uninstall_tinytex (install_tinytex), 3
use_tinytex, 7
use_tinytex (copy_tinytex), 3

xelatex (latexmk), 5