

TinyTeX: A lightweight, cross-platform, and easy-to-maintain LaTeX distribution based on TeX Live

Yihui Xie

Abstract

As a \LaTeX user for 15 years, I have suffered from two problems related to the installation of \LaTeX and maintenance of packages: 1) The full versions of common \LaTeX distributions are often too big, whereas the smaller basic versions often lack packages that I frequently use; 2) It is tedious to figure out which missing packages to install by reading the error log from the \LaTeX compilation. TinyTeX (yihui.name/tinytex/) is my attempt to address these problems. The basic version of TinyTeX is relatively small (150MB on Linux/macOS when installed), and you only install additional packages if/when you actually need them. Further, if you are an R user, the installation of missing packages can be automatic when you compile \LaTeX or R Markdown documents through the R package `tinytex`.

1 Motivation

If you do not want to be bothered by \LaTeX errors that tell you certain class or style files are missing, one way to go is to install the full version of the \LaTeX distribution, which typically contains the vast majority of packages on CTAN. Take \TeX Live for example. The size of its full version is 4 to 5GB. Yes, I do hear the argument that the hard disk storage is fairly cheap today. Why should this 5GB bother us at all? The problems are:

- It can take a long time to download, although we usually do this only once a year. However, if you use a cloud service for continuous integration or testing (e.g., Travis CI) of your software package that depends on \LaTeX , this can be worse, because each time you update your software (e.g., though a GIT commit), the virtual machine or cloud container downloads 5GB again.
- It contains a lot of \LaTeX packages that an average user does not need. I do not know if I'm a representative user, but for the more than 5600 packages on CTAN, I routinely use less than 100 of them. In other words, I'm just wasting my disk space with more than 98% of the packages.
- It takes much longer to update packages if you choose to update all via `tlmgr update --all` (and you will be installing newer versions of packages that you do not need, too).

Without installing the full version, you may be confused when compiling a document and a needed

packages is not installed. The report at github.com/rstudio/rmarkdown/issues/39 is a good example to show how users can be confused. The main reason for the confusion is that an error message like below does not tell you how you could resolve the issue (i.e., which package to install and how to install it):

```
! Error: File 'framed.sty' not found.
```

```
Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: sty)
```

```
Enter file name:
! Emergency stop.
<read *>
```

Even worse, \TeX Live can be different on different platforms. For example, if you use a Linux distribution's packaging of \TeX Live, typically you can not just install the (system) package named `framed` even if you know `framed.sty` is from the (\LaTeX) package `framed`, because \TeX Live is often made available by distributions as *collections* of \LaTeX packages, so you have to figure out which system package contains the \LaTeX package `framed`. Is it `texlive-framed`, or `texlive-latex-extra`? On another front, if you use MacTeX (which is essentially \TeX Live) on macOS, you would usually run `sudo tlmgr install framed`, hence type your password every time you install a package.

Then the next year when a new version of \TeX Live is released, you may have to go through the same pain again: either waste your disk space, or waste your time. One interesting thing I noticed from macOS users was that many of them did not realize that each version of MacTeX was installed to a different directory. For example, the 2018 version is installed under `/usr/local/texlive/2018`, and the 2017 version is under `/usr/local/texlive/2017`. When they started to try TinyTeX (which recommended that they remove their existing \LaTeX distribution), they had realized for the first time that there were five full versions of \TeX Live on their computer, and they were very happy to suddenly regain more than 20GB of disk space.

I wished there were a \LaTeX distribution that contains only packages I actually need, does not require `sudo` to install packages, and is not controlled by system package managers like `apt` or `yum`. I wished there were only one way to manage \LaTeX packages on different platforms. Fortunately, the answer is still \TeX Live, just with a few tricks.

2 The infraonly scheme and the portable mode to the rescue!

There are three possible ways to cut down the size of T_EX Live:

1. Only install the packages you need.
2. Do not install the package source.
3. Do not install the package documentation.

The first way can be achieved by installing a minimal scheme of T_EX Live first, which includes its package manager `tlmgr`, and then install other packages via `tlmgr install`. The minimal scheme is named `scheme-infraonly`, and it is only about 10MB.

The second and third ways can be specified through installation options, which I will mention soon. The package documentation contributes a considerable amount to the total size of a T_EX Live installation. However, I have to admit I rarely read them, and I do not even know where these documentation files are on my computer. When I have a question, I will almost surely end up in a certain post on tex.stackexchange.com, and find a solution there. It is even rarer for me to read the package source files, since I am not a L^AT_EX expert, nor am I interested in becoming an expert.

With the network installer of T_EX Live (tug.org/texlive/acquire-netinstall.html), we can put the above pieces together, and automate the installation through an “installation profile” file. Below is the one that I used for TinyTeX (named `tinytex.profile`):

```
selected_scheme scheme-infraonly
```

```
TEXDIR ./
TEXMFSYSCONFIG ./texmf-config
TEXMFLOCAL ./texmf-local
TEXMFSYSVAR ./texmf-var
```

```
option_doc 0
option_src 0
option_autobackup 0
```

```
portable 1
```

The installation is done through

```
./install-tl -profile=tinytex.profile
```

where `install-tl` is extracted from the Net installer (use `install-tl-windows.bat` on Windows). The full source of the installation scripts can be found on Github at github.com/yihui/tinytex/tree/master/tools. To install TinyTeX on *nix, run `install-unx.sh`; to install it on Windows, run `install-windows.bat`.

I set the `portable` option to 1 above, which means the installation directory will be portable. You can move it anywhere in your system, as long as you know how to handle the `PATH` variable, or call the executables (e.g., `tlmgr` or `pdflatex`) with their full paths. By default, the installation scripts of TinyTeX will try to add T_EX Live’s bin path to the environment variable `PATH`, or create symlinks to a path that is in `PATH` (e.g., `/usr/local/bin` on macOS and `$HOME/bin` on Linux).

A portable installation without admin privileges also means anyone can install and use T_EX Live on any platforms supported by T_EX Live. You can also install a copy to a USB device and use it from there. Users inside an institute no longer need to ask for IT help with managing L^AT_EX packages because of the powerful and useful `tlmgr`. With TinyTeX, `tlmgr` is the one and only way to manage packages directly, and you will not need `sudo`, `apt`, or `yum`.

3 The R package `tinytex`: install missing L^AT_EX packages on-the-fly

Now I only have one last wish for T_EX Live: I wish it could install missing packages on-the-fly like MiKTeX when compiling documents. I do not know how MiKTeX implemented it. I’m primarily an R [2] package developer. I do not know much about the T_EX language or Perl. I know how to search for the package that contains a certain style or class file and install it, e.g.,

```
$ tlmgr search --global --file "/times.sty"
psnfss:
    texmf-dist/tex/latex/psnfss/times.sty
...
$ tlmgr install psnfss
```

I had done this too many times in the past, and thought it might be possible to automate it. I made an attempt in the R package `tinytex` [3]. I guess L^AT_EX experts may frown upon my implementation, but it was the best I could do, given my limited capabilities and knowledge in L^AT_EX.

Basically, I try to compile a L^AT_EX document via an engine like `pdflatex` or `xelatex`, with arguments `-halt-on-error` and `-interaction=batchmode`. If the exit status is non-zero, I will parse the error log and find the error messages. If I made any contribution at all, it would be the following possible error messages that I collected in about a year:

```
! \LaTeX{} Error: File ‘framed.sty’ not found.
/usr/local/bin/mktexpk: line 123: mf:
    command not found
! Font U/psy/m/n/10=psyr at 10.0pt not
loadable: Metric (TFM) file not found
!pdfTeX error: /usr/local/bin/pdflatex
```

```
(file tcrm0700): Font tcrm0700 at 600 not found.
! The font "FandolSong-Regular" cannot be found.
! Package babel Error: Unknown option
'ngerman'. Either you misspelled it
(babel) or the language
definition file ngerman.ldf was not found.
!pdfTeX error: pdflatex (file 8r.enc):
cannot open encoding file for reading
! CTeX fontset 'fandol' is unavailable in
current mode
Package widetext error: Install the
flushend package which is a part of sttools
Package biblatex Info: ... file
'trad-abbrev.bbx' not found
! Package pdftex.def Error: File
'logo-mdpi-eps-converted-to.pdf' not found
```

In the R package `tinytex`, I try to obtain the names of the missing files or fonts or commands (e.g., `framed.sty`, `mf`, `tcrm0700`), run `tlmgr search` to obtain the package name, and `tlmgr install` the package if possible.

The thing that \TeX Live experts may frown upon is that since I do not know all possible missing packages beforehand, I will just keep trying to compile the document, find the missing packages, and install them. In other words, I do not know if there is a missing package unless I actually compile the document and hit an error. If a document contains n missing packages, it may be recompiled n times.

On the bright side, this only needs to be done at most once for a document, so even if it is slow for the first time, the compilation will be much faster next time because all necessary packages have been installed. The process is also automatic (by default), so all you need to do is wait for a short moment. This feature is turned on for R Markdown [1] users, which means if the user's \LaTeX distribution is TinyTeX, they will almost never run into the issue of missing packages when compiling R Markdown to PDF, and the “easy-to-maintain” TinyTeX should not need maintenance at all. As a matter of fact, this article was written in R Markdown, and the first time I compiled it, the `tugboat` package was automatically installed:

```
tlmgr search --file --global /tugboat.cls
tlmgr install tugboat
...
[1/1, ??:??/?:?] install: tugboat [26k]
running mktexlsr ...
done running mktexlsr.
```

The other major thing `tinytex` does is to emulate `latexmk`, i.e., try to compile a \LaTeX document till all cross-references are resolved. The reason to reinvent `latexmk` in an R package is that `latexmk` cannot install missing packages on-the-fly.

To sum it up, if R users compile a \LaTeX document via `tinytex`, usually they will not need to know how many times they need to recompile it, or run into errors due to missing packages. My implementation may be clumsy, but the reaction from users seems to be positive anyway: github.com/yihui/tinytex/issues/7. I hope this could give some inspiration to developers in other communities, and I will be even more excited if \TeX Live adds the native (and professional) support someday, so I can drop my poor implementation.

4 Discussion

There is no free lunch. TinyTeX also has its drawbacks, and you have to consider whether they matter to you. First of all, when installing TinyTeX, you are always installing the very latest version of \TeX Live. However, as I have mentioned, TinyTeX is a portable folder, so you can save a copy of a certain version that worked for you, and use it in the future.

Secondly, the installation of TinyTeX and the (automatic) installation of additional \LaTeX packages requires an Internet connection. This may be the biggest drawback of TinyTeX. If you plan to work offline, you will have to make sure all packages have been installed in advance.

Thirdly, TinyTeX was created mainly for individual users who install TinyTeX for themselves. If a sysadmin wants to install a shared copy of TinyTeX for multiple users, there will be more technical details to learn (in particular, issues related to permissions, the “user mode”, and packages that are not “relocatable”). I have mentioned them on the FAQ page: yihui.name/tinytex/faq/.

Lastly, TinyTeX is essentially a version of \TeX Live installed through an installation script. I did not provide prebuilt binaries, even though it would be easy technically. I do not fully understand the \TeX Live license and \LaTeX package licenses, but I guess I would be very likely to violate these licenses if I provide binaries without also shipping the source files inside at the same time. Anyway, installing TinyTeX over the Internet usually takes only a minute or two, so this may not be a big concern.

I hope you might find TinyTeX (and the R package `tinytex`, if you happen to be an R user, too) useful. If you have any feedback or questions or bug reports, please feel free to post them to the Github repository: github.com/yihui/tinytex.

References

- [1] J. Allaire, Y. Xie, et al. *rmarkdown: Dynamic Documents for R*, 2019. R package version 1.14. <https://CRAN.R-project.org/package=rmarkdown>

`rmarkdown`

- [2] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. <https://www.R-project.org/>
- [3] Y. Xie. *tinytex: Helper Functions to Install and Maintain 'TeX Live', and Compile 'LaTeX' Documents*, 2019. R package version 0.14. <https://CRAN.R-project.org/package=tinytex>

◇ Yihui Xie
RStudio, Inc.
`xie@yihui.name`
<https://yihui.name>