

The native TeX Live integration

TeX Live is nowadays the last maintained TeX distribution. Due to its size, it is not provided in the Slackware's official repository. That's why many (most of?) people use the dedicated SlackBuild from [SlackBuilds.org](https://slackbuilds.org) to get it as a huge complete monolithic package.

Of course, this solution is a good one, as everybody will then get a perfectly integrated package, able to feed far more needs than you would look for. Nevertheless, as always there are two faces on the coin. First, it's huge, *really* huge, easily more than a whole customized system. Second, it's not flexible: once it's installed, you can't update its contents, which means you are stuck with the potential bugs you may encounter until the next major release, possibly for a whole year.

That's why this tutorial will try to show you how to obtain a more updatable and customizable installation, using the native upstream TeX Live distribution. The idea is to put it in its dedicated directory, then manage it with its own tools. Of course it will not be as proper as an orthodox package, but, we will see, not as dirty as it sounds first.

Prepare the TeX Live sanctuary

Here, we will make a place for the future TeX Live installation. The first thing we need is someone to manage it, a user which will be the only one to deal with the TeX package repositories. We could of course use `root` for this, but this is not a good idea, because granting the right to write everywhere to a third party system is not secure at all. That's why we'll create a dedicated system user (ie. a user without any shell to login):

```
# useradd -rU -c "The TeX Live user" -d / -s /bin/false tuser
```

Right. Now, we make the unique place where our new `tuser` will be allowed to write (YYYY refers to the TeX Live's version, which is the year each one is released, replace it with the one you aim):

```
# mkdir /opt/texlive-YYYY/
# chown tuser:tuser /opt/texlive-YYYY/
# ln -sf texlive-YYYY /opt/texlive
```

The last command creates a generic symlink, which will be useful when upgrading the installation, allowing the old release to temporarily stand beside the new.

Installed here though, our TeX Live will be almost useless, because no one will have it in his `PATH`. We therefore add the two suitable scripts into `/etc/profile.d`, respectively named `texlive.sh` and `texlive.csh`:

[texlive.sh](#)

```
#!/bin/sh
if [ ! "$(id -u)" = "0" ]; then
    PATH=$(echo $PATH | sed 's#:\.\.?###s###:/opt/texlive/bin/i386-
linux:.#')
```

```
export PATH
fi
```

[texlive.csh](#)

```
#!/bin/csh
if ( `id -u` != "0" ) then
    setenv PATH `echo $PATH | sed 's#:\.\/?##;s##:/opt/texlive/bin/i386-
linux:.#'`
endif
```

As you can see, in both we insert the new directory at the end of the PATH. This is because we want to use our system's binaries rather than the ones shipped by the TeX live distribution.

You've maybe also noticed this addition won't be made to the root's PATH. It's once again normal, because root is not intended to execute these binaries, we want to use him only to administer our TeX Live installation. More precisely, we want him to use the `tluser` we created to perform this. That's why the only thing root should have in his path is a wrapper for `tlmgr`, the native TeX Live manager. Here it is:

[tlmgr](#)

```
#!/bin/sh

if [ $(id -u) -ne 0 ]; then
    echo "(E) Only root can use this script." >&2
    exit 1
fi

# tluser must use the TeX Live binaries rather than the system's ones.
export PATH=/opt/texlive/bin/i386-linux:/bin:/usr/bin
exec sudo -u tluser tlmgr "$@"
```

We put it in `/usr/sbin`, then we ensure it is executable:

```
# chmod 755 /usr/sbin/tlmgr
```

This completes the last step to setup our sanctuary. Now let's see how to get the lion into it.

Install the native TeX Live

As you probably don't plan to install all of its components, it is better to perform a network installation. [Get the Unix network installer tarball](#), untar it, enter the directory and run it **as `tluser`** (this last step requiring to be root):

```
$ tar xf install-tl-unx.tar.gz -C /tmp
```

```
$ cd /tmp/install-tl-YYYYMMDD
$ su
Password:
# sudo -u tuser ./install-tl
```

A text menu will be displayed, providing configuration items you select by passing letters or numbers. The only required setting is the final installation directory: enter the `directories` section then specify our sanctuary (`/opt/texlive-YYYY`) as `TEXDIR`. Beyond, you're the cook! Personally, I select the basic scheme in `installation scheme`, deselect all the languages except mine in `language collections`, and disable source and doc tree installation in `option`. Once I've installed the few packages and fonts I need over this, I get a ~80M big TeX Live installation. If you nevertheless prefer a more complete thing, you can pick up in `installation scheme` a replacement for the teTeX historically provided by Slackware. Once you are done, order to `start installation to hard disk`.

Manage the native TeX Live

It is time to learn a little bit more about `tlmgr`, the *TeX Live Manager*. This section is just a kick-starter explaining how to perform the more common tasks, though. To get a complete description, read `tlmgr --help`.

The first thing you may want to have is a list of the installed packages. Try:

```
$ tlmgr info --only-installed
```

To get an extension you miss, first identify the package it belongs to (here we look for the `fullpage` extension):

```
$ tlmgr search --global --file fullpage
tlmgr: package repository http://...
context-fullpage:
  texmf-dist/doc/context/third/fullpage/README
  texmf-dist/doc/context/third/fullpage/fullpage-doc.pdf
  texmf-dist/tex/context/interface/third/t-fullpage.xml
  texmf-dist/tex/context/third/fullpage/t-fullpage.mkii
  texmf-dist/tex/context/third/fullpage/t-fullpage.mkiv
context-gnuplot:
  texmf-dist/doc/context/third/gnuplot/fullpage-example.pdf
  texmf-dist/doc/context/third/gnuplot/fullpage-example.tex
preprint:
  texmf-dist/doc/latex/preprint/fullpage.pdf
  texmf-dist/tex/latex/preprint/fullpage.sty
  texmf-dist/source/latex/preprint/fullpage.drv
  texmf-dist/source/latex/preprint/fullpage.dtx
  texmf-dist/source/latex/preprint/fullpage.ins
```

Then, once you've found (in our example, we need the `preprint` package), install it (remember you have to be root to add or modify anything):

```
# tlmgr install preprint
```

To make a general update, first check for the potential ones:

```
$ tlmgr update --list
```

If there are some, order to first update `tlmgr` itself (it's always better), then to perform the general updating of the packages:

```
# tlmgr update --self --all
```

In the case you don't want to update all the packages, replace the `--all` option with the name(s) of the package(s) you aim. Last, removing a package is trivially done with:

```
# tlmgr remove <package>
```

As you can see, there's not much to know in order to deal with a native TeX Live installation, and it allows to really precisely specify what should be done.

Getting it packaged however

Let's rather call this a *snapshot* then, a package you will keep for you, to perform quick re-installations (keep in mind it doesn't make sense to distribute it as long as it ships `tlmgr`). Assuming you've rigourously followed the instructions given all along this tutorial, here is a `tlshot` named script, which should do the job:

[tlshot](#)

```
#!/bin/sh
#
# TlSHOT - make a package from the current native TeX Live
# installation.
#
# Copyright (c) 2012, Sébastien Boillod <sbb at tuxfamily dot org>.
#
# Permission to use copy, modify, and/or distribute this software for
# any
# purpose with or without fee is hereby granted, provided that the
# above
# copyright notice and this permission notice appear in all copies.
#
# THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL
# WARRANTIES
# WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE
# FOR ANY
# SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
```



```
fi
rm -rf "$root"

# EoF
```

The result should be a package in /tmp (or the directory you've specified through the TMPDIR environment variable), named `texlive-YYYYX-i386-1.txz`, where YYYY is as usual the TeX Live's version, and X is a lower letter dynamically attributed to distinguish a snapshot from another. The TeX Live directory packaged is the one pointed by the `/opt/texlive` symlink.

If you choose to use snapshots, you'll probably want to disable the automatic backups performed by `tlmgr` (it won't be very useful, while consuming space and making the snapshots bigger):

```
# tlmgr option autobackup 0
```

This time, we have reached the end of this tutorial. I hope you've found something useful here, at least as a basis for your own cleverer solutions. If what you've read so far has globally convinced you, it should not require too much work to tune it according to your tastes. So, play with it and find your perfect TeX Live installation!

Sources

- Originally written by [Sébastien Boillot](#) for the Slackware Documentation Project.

[howtos](#), [texlive](#), [LaTeX](#), [author vaporseb](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
<https://docs.slackware.com/es:howtos:software:texlive>

Last update: **2019/02/28 01:54 (UTC)**

