

TeX Live 2008 and the TeX Live Manager*

Abstract

TeX Live 2008 has been released recently, and the DVDs are ready to go gold. This is the first release of TeX Live shipping the TeX Live Manager, `tlmgr` for short. Besides taking over some of the tasks from `texconfig` (which has never been available for Windows) it finally brings many new features to the TeX Live world, most importantly the option for dynamic updates.

This article will present the new TeX Live Installer, the TeX Live Manager, and at the end lists other changes in TeX Live 2008.

Important note

This article describes the status of the TeX Live Manager as it will be shipped around October 2008, and not the one on DVD. The version on the DVD works fine for local configuration tasks (which is why we felt it could be shipped), but is not sufficiently robust for reliable updates over the Internet. Users' first update will be to get the new `tlmgr`.

Introduction

After more than one year of development work TeX Live 2008 has been released with a complete new infrastructure [?]. At first these infrastructure changes were only relevant for the developers themselves, since it made life (a bit) easier and the system more consistent due to the elimination of duplicated information.

As a first user-visible change came the unification of the installer, so that all supported platforms now share the same installer. Furthermore, this installer has gotten a GUI which also is uniform across all platforms. On Unix systems the only prerequisites are a Perl installation, and for the GUI the installation of Perl/Tk. On Windows we ship a minimal Perl with the necessary modules.

The first part of this article will give an overview of the new installer.

The second user-visible change came from the addition of the TeX Live Manager, or `tlmgr` for short, to the list of programs. It manages an existing TeX Live installation, both packages and options. Besides performing many of the same actions as `texconfig` it has the ability to install additional packages, update and remove existing ones, make backups, search within and list all packages.

The new installer

The creation of a new TeX Live installer was necessitated by the new package infrastructure [?]. From a user's point of view the new installer has only one visual change, but there other significant changes. In particular:

- It is possible to install TeX Live from the Internet.

*Originally presented at the GuIT Conference 2008 in Pisa, and published in *ArsTeXnica*, issue 6

```

=====> TeX Live installation procedure <=====
==> Note: Letters/digits in <angle brackets> indicate menu items <==
==>         for commands or configurable options           <==

Proposed platform: Intel x86_64 with GNU/Linux

<B> binary systems: 1 out of 15

<S> Installation scheme (scheme-full)

Customizing installation scheme:
  <C> standard collections
  <L> language collections
  83 collections out of 84, disk space required: 1426 MB

<D> directories:
  TEXDIR (the main TeX directory):
    /usr/local/texlive/2008
  TEXMFLOCAL (directory for site-wide local files):
    /usr/local/texlive/texmf-local
  TEXMFSYSVAR (directory for variable and automatically generated data):
    /usr/local/texlive/2008/texmf-var
  TEXMFSYSCONFIG (directory for local config):
    /usr/local/texlive/2008/texmf-config
  TEXMFHOME (directory for user-specific files):
    ~/texmf

<O> options:
  [ ] use letter size instead of A4 by default
  [X] create all format files
  [X] install macro/font doc tree
  [X] install macro/font source tree
  [ ] create symlinks in standard directories

<V> set up for running from DVD

Other options:
=====
<I> start installation to hard disk
<H> help
<Q> quit

Enter command:

```

Figure 1. Main menu of the text mode installer

- There is just one installer, which can run either in text mode, emulating the former `install-tl.sh` shell script, or in GUI mode (more or less emulating the TeX Live 2007 `tlpmgui`).
- The Windows installation is much closer to using the same implementation as Unix.

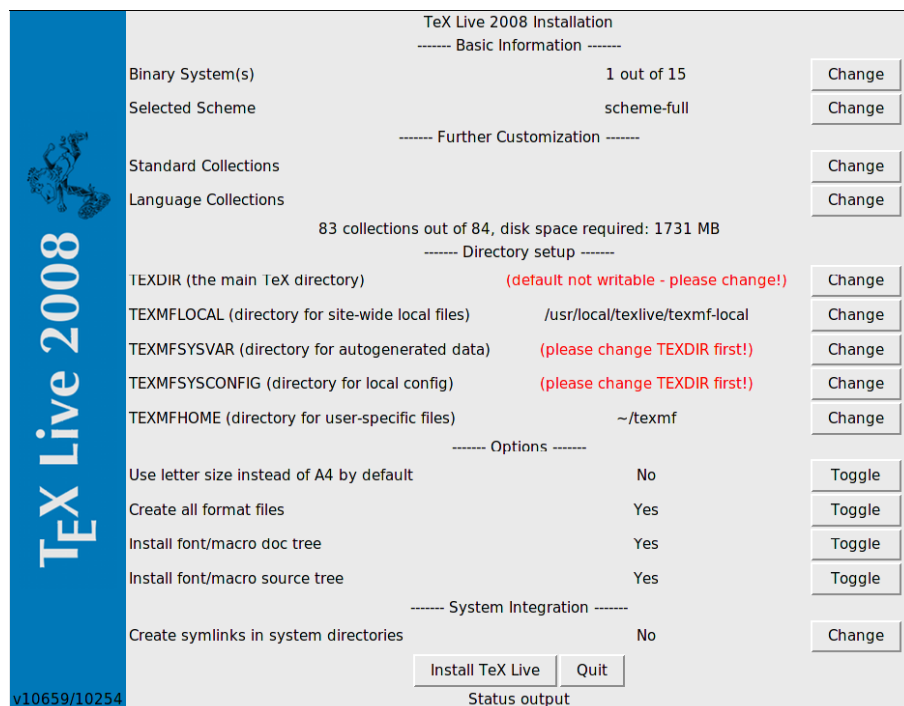


Figure 2. Main window of the GUI installer

Install T_EX Live from the Internet

If you got (by the time you read this article) a T_EX Live DVD you can just start the installer as usual. On Windows this will be by default the GUI installer (see below), on Unix the text mode installer.

We also ship an installation package [?] containing all the necessary files for an installation over the network. By default, normal installation will use a CTAN mirror, selected using the <http://mirror.ctan.org> service. (See <http://tug.org/2cctan.html#sites>.)

Two installers for network downloads are provided. `install-tl-unx.tar.gz` supports Unix only. `install-tl.zip` additionally contains a small subset of Perl for Windows which is required to bootstrap the system. The latter works on all platforms supported by T_EX Live. The sole reason for providing a separate package for Unix is its significantly smaller size.

In any case you can override the source from which you want to install with the command line option `-location`.

The text mode installer

If you have used T_EX Live in recent years you will see no big changes in the text mode installer (see fig. 1); we tried to keep it as close as possible to the one used in former releases. One new option is at the bottom of the menu, namely *set up for running from DVD*. This is what we call *live installation*: it sets up a minimal writable environment on your computer, while all the input files and binaries remain on the DVD.

The GUI Installer

The GUI installer has nearly the same functionality as the text version; the option to set up for live installation is the only missing piece. It is written in Perl/Tk and thus should run on all platforms (on Unix Perl/Tk has to be installed).

The main window can be seen in fig. 2. It should remind you very much of the text mode installer. As with the text mode installer it allows you to change which binary systems should be installed (fig. 3), select the scheme to be installed (fig. 4), where a *scheme* is a pre-defined set of collections to be installed, and further specify in more detail which collections (a collection is a set of packages) and which language packages to install (fig. 5 and 6). You can select the installation directories for TeX Live and toggle some options, all in line with the installer of recent years and the text mode installer.

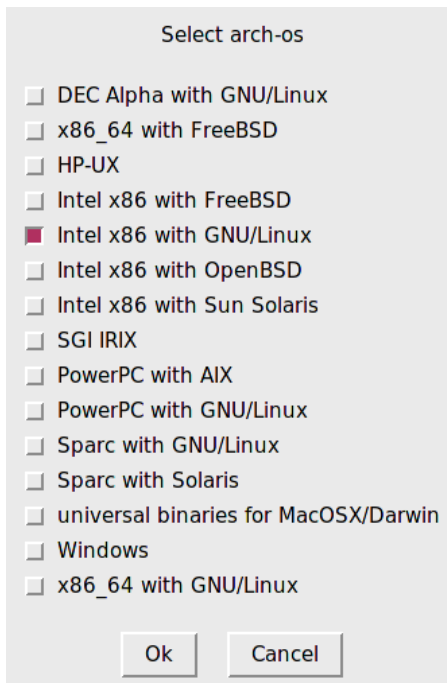


Figure 3. Binary system select window

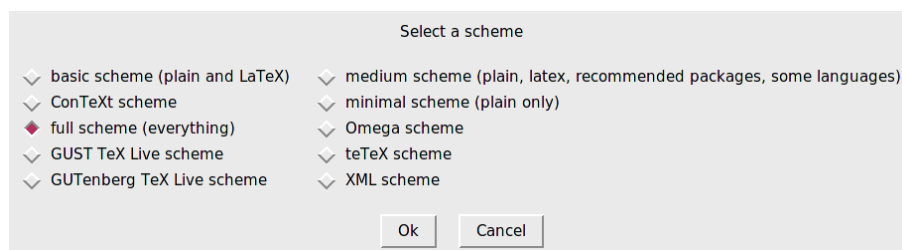


Figure 4. Scheme select window

During installation the main window's status line will indicate what is going on currently, and at the same time the program will print out to the terminal the same output as the normal installer.

For both the text mode installer and the GUI mode installer, a log file with more details is created in the installation directory as `install-tl.log`. (If you report installation problems, please send us this log file.)

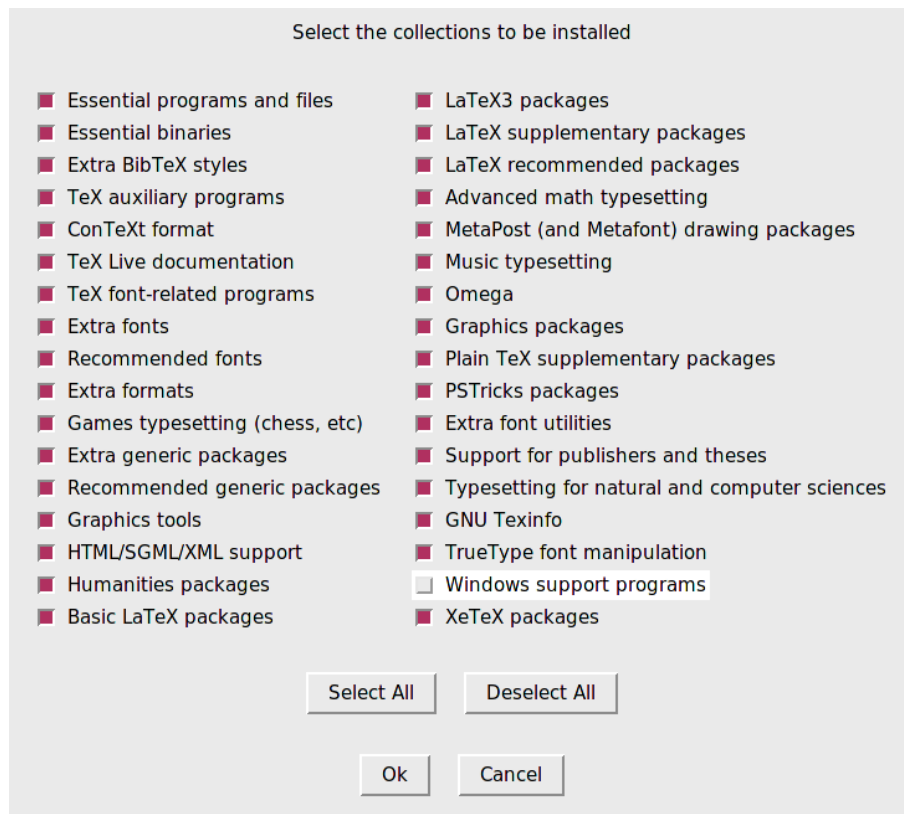


Figure 5. Collections select window

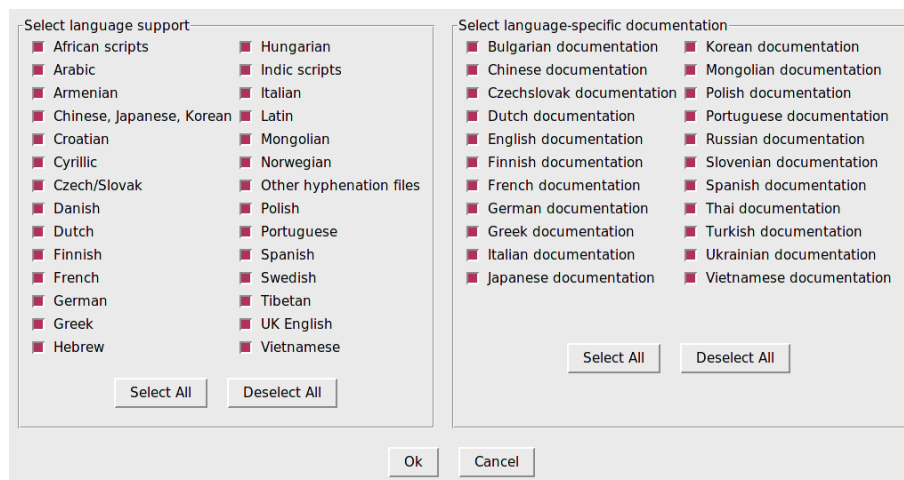


Figure 6. Language packs select window

Bringing Windows in line with Unix

T_EX Live 2008 supports Windows 2000 and later. By dropping older Windows versions, there is much less need to treat Windows specially.

Under Windows 2000 and later, users have a real home directory, viz. `%USERPROFILE%`, usually `C:\Documents and Settings\username`.

This is now reflected in tilde expansion by Kpathsea: `~/texmf` is expanded to `%USERPROFILE%\texmf` under Windows and to `$HOME/texmf` under Unix.

It is also possible to differentiate between system settings and user settings. Happily, there is no longer any need to have a different set of `texmf` trees on Windows, or to leave out scripts such as `fmtutil-sys` and `updmap-sys`. We also have a single `texmf.cnf` used on all platforms.

The T_EX Live Manager

T_EX Live Manager provides a wealth of options and commands; we will explain them all here, some tersely, some in more detail. Of course we expect to add more features in the future.

The T_EX Live Database

First of all, it is important to understand where all the information about installed packages and other options are saved. This is the T_EX Live database, which normally can be found in `ROOT/tlpkg/texlive.tlpdb` (where `ROOT` is the destination folder you have chosen at installation time). It contains the list of all packages, all the files installed, and in addition to that collects configuration information like the default installation source, and the options you have set at installation time (e.g., whether you want A4 or letter-size paper by default).

Most of `tlmgr`'s actions will load the local database, and many actions will also load a remote database: If you want to install a package, the T_EX Live Manager will load the database of the specified installation server and checks whether this package exists.

Although we say *remote*, it is not necessarily a remote network location. If you install from DVD the default installation source will be the DVD, and `tlmgr` will load the database located on the DVD when needed.

General syntax of `tlmgr`

The general syntax of `tlmgr` is

```
tlmgr [opt]... action [opt]... [arg]...
```

The first set of options before the main `action` configure the general operation of `tlmgr`, while the second set of options are specific to the chosen `action`. We do not support mixing and reshuffling of all these options, partly for the sake of clarity, and also for programming reasons. The first set of options can contain:

`--location loc` specifies the location from which packages should be installed or updated, overriding the location found in the installation's T_EX Live Package Database (TLPDB).

`--gui` starts the GUI version of `tlmgr`. The GUI does not support all the bells and whistles of the command-line program. It is in fact a separate program that calls the command-line version to do the actual work. The difference between this `--gui` option and the `gui` action (see below) is that given the option, `tlmgr` tries to open the GUI directly at the screen for the specified `action`.

`--gui-lang ll` selects the language in which the GUI will appear. Normally the GUI tries to deduce your language from the environment (on Windows via the registry, on Unix via `LC_MESSAGES`). If that fails you can select a different language by giving this option a two-letter language code.

Furthermore, some standard options are supported: `--help` (also `-h`) for getting help, `-q` which suppresses informational messages, and `-v` (verbose) for turning on

debugging options. With `--version` the script will show you the version of your T_EX Live system and of itself.

The actions

There is a (permanently growing) list of actions, currently: `help`, `version`, `gui`, `install`, `update`, `backup`, `restore`, `remove`, `option`, `paper`, `arch`, `search`, `show`, `list`, `check`, `uninstall`, `generate`.

The general actions.

- `search` [*option* . . .] *what* Without any options, search in the list of locally installed packages for package names or descriptions matching *what*. If you give the option `--global` it also searches the remote database. This might differ in case you have not installed all of T_EX Live, but only a part of it. Finally, if you specify `--file` then files are searched, and not package names.
- `show` *pkg* . . . gives you more detailed information on the listed packages. If all packages are installed locally, it does not consult the remote database.
- `list` [*collections*|*schemes*] With no argument, lists all packages available at the default install location, prefixing those already installed with "i ". With an argument lists only collections or schemes, as requested.
- `uninstall` This action will ask for confirmation, and then remove the entire installation. Don't do it or we will be sad. If you give the `--force` option, it does not even ask, but proceeds immediately with the removal.
- `check` [*files*|*collections*|*all*] Executes one (or all) check(s) on the consistency of the installation. For *files* it checks that all files listed in the local database are actually present, and lists those missing. The option `--use-svn` will use the `svn` command to check for the files.
- `gui` starts the GUI, as explained above at `--gui`.
- `version` is the same as `--version`.
- `help` is the same as `--help`.

The configuration actions.

- `option` [*show*] Shows all configuration settings currently saved in the local database. The *show* option is accepted and ignored.
- `option` *key* [*value*] Without the *value*, shows the current value of the configuration option *key*; with *value*, sets this configuration option. Currently accepted keys are `location` (default installation location), `formats` (create formats at installation time), `docfiles` (install documentation files), `srcfiles` (install source files). These are the options you have set at installation time and will be honoured at *later* install and upgrade actions. For example, changing the `docfiles` options from `false` to `true` will not install or remove the already present documentation files. But a subsequent update will install or remove them.
- `paper` *paper* Sets the default papersize; possible values are `a4` and `letter`.
- `program` *paper* [*help*|*paper*] This allows setting different paper sizes for the specified *program*: `xdvi`, `dvips`, `pdftex`, `dvipdfm`, `dvipdfmx`, `context`. Without any additional argument it reports the currently selected papersize. With `help`, it issues all the supported paper sizes for that program. And if you specify a paper size, it will be set as default papersize for the given program.

- `generate what` This command generates one or more configuration files, as follows:

Giving `language.dat` for `what` generates the `language.dat` file which specifies the hyphenation patterns to be loaded for LaTeX-based formats. Giving `language.def` for `what` generates the `language.def` file which specifies hyphenation patterns to be loaded for etex-based formats. Specifying `language` for `what` generates both of these files.

Giving `fmtutil` for `what` generates the `fmtutil.cnf` file which contains the definitions of all formats available.

Giving `updmap` for `what` generates the `updmap.cfg` file which lists all the installed font map files.

For `fmtutil` and the language files, recreating is normal and both the installer and `tlmgr` routinely call that.

For `updmap`, however, neither the installer nor `tlmgr` use `generate`, because the result would be to disable all maps which have been manually installed via `updmap-sys --enable`, e.g., for proprietary or local fonts. Only the changes in the `--localcfg` file mentioned below are incorporated by `generate`.

On the other hand, if you only use the fonts and font packages within TeX Live, there is nothing wrong with using `generate updmap`. Indeed, we use it to generate the `updmap.cfg` file that is maintained in the live source repository.

If the files `language-local.dat`, `language-local.def`, `fmtutil-local.cnf`, or `updmap-local.cfg` are present under `TEXMFLOCAL` in the respective directories, their contents will be simply merged into the final files, with no error checking of any kind.

The package management actions.

- `install pkg...` installs the packages given as argument. By default, installing a package also installs all of its dependencies. The following options are supported: `--no-depends` will not install dependent packages. There is also `--no-depends-at-all` which in addition disregards the tightly coupled packages architecture-specific executables; for example, `bin-bibtex` and `bin-bibtex.i386-linux`. That is something you should never use unless you are sure you know what you are doing. `--dry-run` fakes the installation without changing anything.
- `update pkg...` updates the packages given as arguments. In addition, if the `pkg` is a collection, and the remote server has new packages in this collection, they will be installed, following the dependencies specified in the collection. Options:
 - `--list` Lists the packages which would be updated or newly installed, but does not do the update. It also lists the revision numbers of the local and the remote packages.
 - `--all` Update all out-of-date packages.
 - `--dry-run` Fake the updates without changing anything.
 - `--backupdir directory` Save a snapshot of the current package (as installed) in `directory`, before the package is updated. This way one can easily recover in case an update turned out as not working. See the `restore` action for details.
 - `--no-depends` Do not install normal dependencies.
 - `--no-depends-at-all` See `install` above for this option.

- `remove pkg...` removes the packages given as arguments. Removing a collection will remove all package dependencies (but not collection dependencies) in that collection, unless `--no-depend`s is specified. However, when removing a package, dependencies are never removed.

Options:

`--no-depend`s Do not remove dependent packages.

`--no-depend`s-`at-all` See `install` above for this option.

`--force` By default, when removing a package or collection would invalidate a dependency of another collection/scheme, the package is not be removed and an error is given. With this option, the package will be removed unconditionally. Use with care.

`--dry-run` Fake the removals without actually changing anything.

- `backup pkg...` makes a backup of the given packages, or all packages with `--all`, to the directory specified with `--backupdir` (must exist and be writable).

The following options are supported:

`--backupdir directory` The directory is a required argument and must specify an existing directory where backups are to be placed.

`--all` Make a backup of all packages in the T_EX Live installation. This will take quite a lot of space and time.

- `restore --backupdir dir [pkg [rev]]`

If no `pkg` is given (and thus no `rev`), lists the available backup revisions for all packages.

With `pkg` given but no `rev`, list all available backup revisions of `pkg`.

With both `pkg` and `rev`, tries to restore the package from the specified backup.

The option `--backupdir dir` is required, and must specify a directory with backups.

The option `--dry-run` is also supported, as usual.

- `arch operation arg...` If `operation` is `list`, this lists the names of architectures (i386-linux, ...) available at the default install location.

If `operation` is `add`, adds the executables for each of the following arguments (architecture names) to the installation.

The option `--dry-run` is also supported, as usual.

Typical usage of `tlmgr`

Here we present some typical usage examples of the T_EX Live Manager.

Installing a new collection. Suppose that you installed `scheme-medium` and then realize that the hyphenation patterns for some language you are using haven't been installed, say, Norwegian. First you fire up `tlmgr` to search for the support:

```
$ tlmgr search --global norwegian
collection-langnorwegian - Norwegian
hyphen-norwegian -
```

and then to install this collection:

```
$ tlmgr install collection-langnorwegian
install: collection-langnorwegian
install: hyphen-norwegian
regenerating language.dat
regenerating language.def
```

and then it continues to regenerate all the format files depending on either `lang2` `Guage.dat` or `language.def`. (If the `formats` option is changed to `false` in the local database, the format rebuilding will be skipped. The default is to do so, to keep them up to date without manual intervention.)

Searching for a package. You want to typeset an invitation in a special form, say in the shape of a heart. Your first try is

```
$ tlmgr search paragraph
```

but that yields no output. Maybe it's not installed? So try a global search:

```
$ tlmgr search -global paragraph
tlmgr: installation location /src/TeX/texlive-svn/Master
bigfoot - Footnotes for critical editions
edmargin - Multiple series of endnotes for critical editions
footmisc - A range of footnote options
genmpage - Generalization of LaTeX's minipages
hanging - Hanging paragraphs
ibycus-babel - Use the Ibycus 4 Greek font with Babel
insbox - A TeX macro for inserting pictures/boxes into paragraphs
layouts - Display various elements of a document's layout
lettrine - Typeset dropped capitals
lineno - Line numbers on paragraphs
lipsum - Easy access to the Lorem Ipsum dummy text
moresize - Allows font sizes up to 35.83pt
ncctools - A collection of general packages for LaTeX
paralist - Enumerate and itemize within paragraphs
picinpar - Insert pictures into paragraphs
plari - Typesetting stageplay scripts
seqsplit - Split long sequences of characters in a neutral way
shapepar - A macro to typeset paragraphs in specific shapes
vwcol - Variable-width multiple text columns
```

and here we are, `shapepar` seems to be what's needed. So let us see what it is:

```
$ tlmgr show shapepar
tlmgr: installation location /src/TeX/texlive-svn/Master
Package: shapepar
Category: Package
ShortDesc: A macro to typeset paragraphs in specific shapes.
LongDesc: \shapepar is a macro to typeset paragraphs in a
special ...
Installed: No
Collection:collection-latexextra
```

Ok, confirmed, now we can either install the respective collection using

```
$ tlmgr install collection-latexextra
```

which will install quite a lot of packages, or only that one single package in the hope that it does not depend on anything else:

```
$ tlmgr install shapepar
tlmgr: installation location /src/TeX/texlive-svn/Master
install: shapepar
running mktexlsr
...
```

These examples are about finding uninstalled packages. The default for \TeX Live is a full installation, i.e., everything is installed that is available.

Updating your installation. After the initial installation you want to get the latest and greatest of everything, but first you want to see what that means:

```
$ tlmgr update --list
tlmgr: installation location /mnt/cdrom
Cannot load TeX Live database from /mnt/cdrom at /home/norbert/tltest )
  ↵ /2008/bin/i386-linux/tlmgr line 1505, <TMP> line 1982.
```

Hmm, there seems to be an error, it tries to install from the DVD which you returned to your friend last week. Well, then you should switch to the network installation source; best to do it for all future sessions by saving it as default location. But what was that strange address again? Fortunately you can tell tlmgr to use CTAN and it will know what to do:

```
$ tlmgr option location ctan
tlmgr: setting default installation location to http://mirror.ctan.org )
  ↵ /systems/texlive/tlnet/2008
```

Fine. Now let us see what we can upgrade:

```
$ tlmgr update --list
shapepar: local: 10400, source: 10567
bin-texlive: local: 10693, source: 10750
pdftex: local: 10622, source: 10705
texlive.infra: local: 10685, source: 10748
```

Well, some things are there, so let us update all of them at once:

```
$ tlmgr update --all
update: shapepar (10400 -> 10567) ... done
update: bin-texlive (10693 -> 10750) ... done
update: pdftex (10622 -> 10705) ... done
update: texlive.infra (10685 -> 10748) ... done
running mktexlsr ...
```

Paper size configuration. You are moving to Japan and want letter as your default paper size; nothing easier:

```
$ tlmgr paper letter
```

will switch to letter for the most important programs, and at also recreate the formats.

The GUI for tlmgr

To make most Windows users and some Unix users happy we provide a front end for the T_EX Live Manager written in Perl/Tk. It does not do the actual work, but leaves that for tlmgr. It also does not provide quite the full functionality of tlmgr, but almost all of it is there.

This program features several screens for different functionalities: installation, update, and removal of packages, removal of T_EX Live as a whole, architecture support and configuration.

The GUI is started with either `tlmgr gui` or `tlmgr --gui action` where `action` is one of the actions given above. In the latter case it tries to open the respective screen of the GUI.

The install screen

The first window to be seen normally is the package installation screen (fig. 7).

At the top you see the current installation source, as given either on the command line of tlmgr, or in the absence of a command line argument as taken from

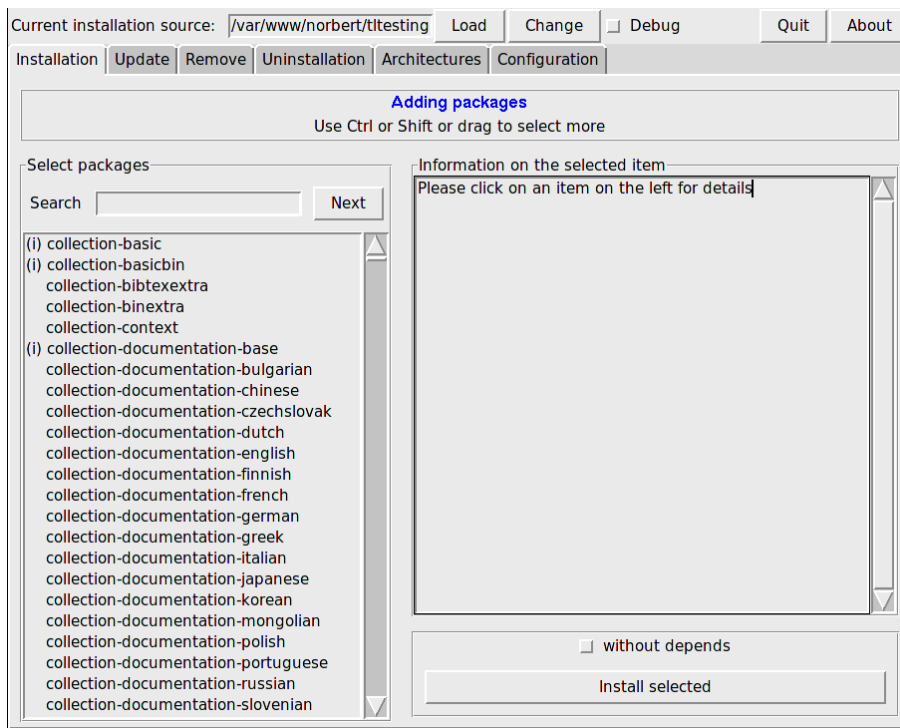


Figure 7. T&E Live Manager GUI install screen

the default option. It is *not* loaded automatically, you have to press the Load button, or the Change button to select a different installation source for this run only. Below you see the list of available packages on the left, first all the collections and schemes, then all the other packages in alphabetic order. You can search by entering a string into the search text field, which immediately jumps to the first entry. The button Next jumps to the next match. After selecting one package you can see its description in the right half of this screen. Below there is the action button for installing the selected package(s), and also a switch that allows you to install a package without those it depends on.

The update screen

The update screen is similar to the install screen, but only lists those packages which have an upgrade available on the installation location. The upper part of the right pane gives you information on the package, and in the action area below you see two buttons, one for updating only the selected packages, and one for updating all packages.

In fig. 8 you can see the update screen with updates available and the information for the selected package shown in the right part of the screen.

The remove screen

The remove screen is also similar to the install screen, with the list of all installed packages in the left part, the information window in the upper right part, and the action area with two toggles and the remove button in the lower right part; see fig. 9.

The two toggles correspond to the option `--force` and `--no-depends` of the `tlmgr remove` action, see above.

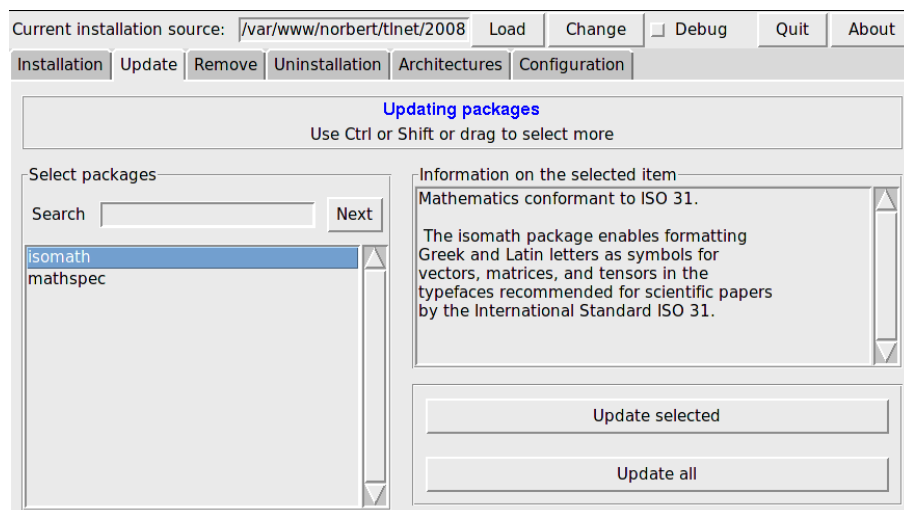


Figure 8. T_EX Live Manager GUI update screen

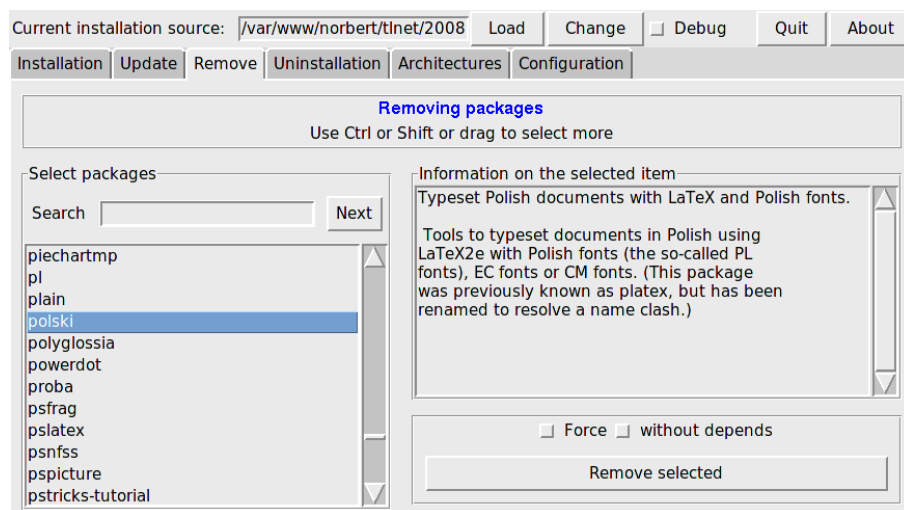


Figure 9. T_EX Live Manager GUI remove screen

The uninstallation screen

This screen only sports one button which allows you to completely remove the T_EX Live installation from your system. This button is not present on Windows systems, being replaced by an informational note that you should use the Add/Remove entry from the Control Panel.

The architectures screen

T_EX Live allows you to install the binaries for several architecture-operating system combinations in case you want to distribute your installation via NFS or other means in an inhomogen local network, see fig. 10.

This screen lists the available architectures at the current installation source, and allows you to select new architectures to be installed by pressing the Apply changes button.

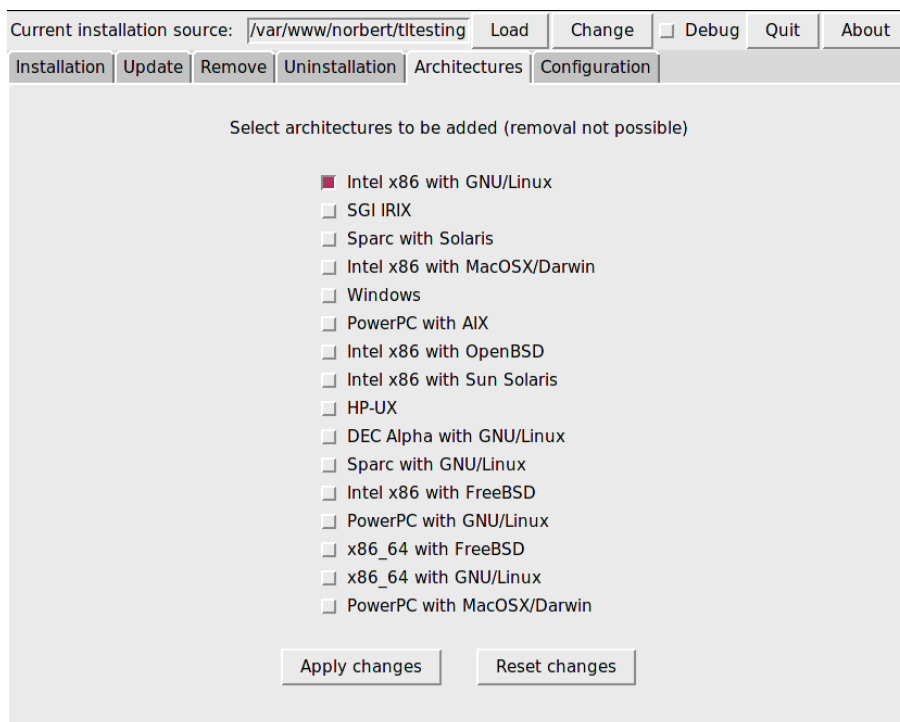


Figure 10. TeX Live Manager GUI architectures screen

Note that the *removal* of architectures is currently not supported, and that the whole screen is disabled on Windows systems since Windows does not support normal symbolic links.

The config screen

This screen allows the user to comfortably examine and set the various options of the TeX Live installation, see figure 11.

In the upper part you can change the defaults for the installation source, whether formats should be created (and updated) by default, and whether macro/font documentation and source files should be installed.

In the lower left part you can set the letter for all the programs to either A4 or letter, or for each program individually. In the latter case you can choose from a wide range of paper formats depending on the programs support.

In the lower right part there are some convenience buttons for updating the `ls-R` databases, the outline font list (`updmap-sys`) and rebuilding all formats.

Execution of the commands

As mentioned above, this GUI is only a front end and leaves the actual work to `tlmgr` itself. So every action you do (installation, removal, etc.) will pop up a window where the output of the `tlmgr` process is shown.

On Unix systems that output will be shown immediately. Windows lacks good support for forking in Perl/Tk, and thus you have to wait until the whole process has terminated before the output appears. That can take quite some time, so please be patient.

We are working on merging the `tlmgr` and its GUI into one program so that the output would become more immediate in all cases.

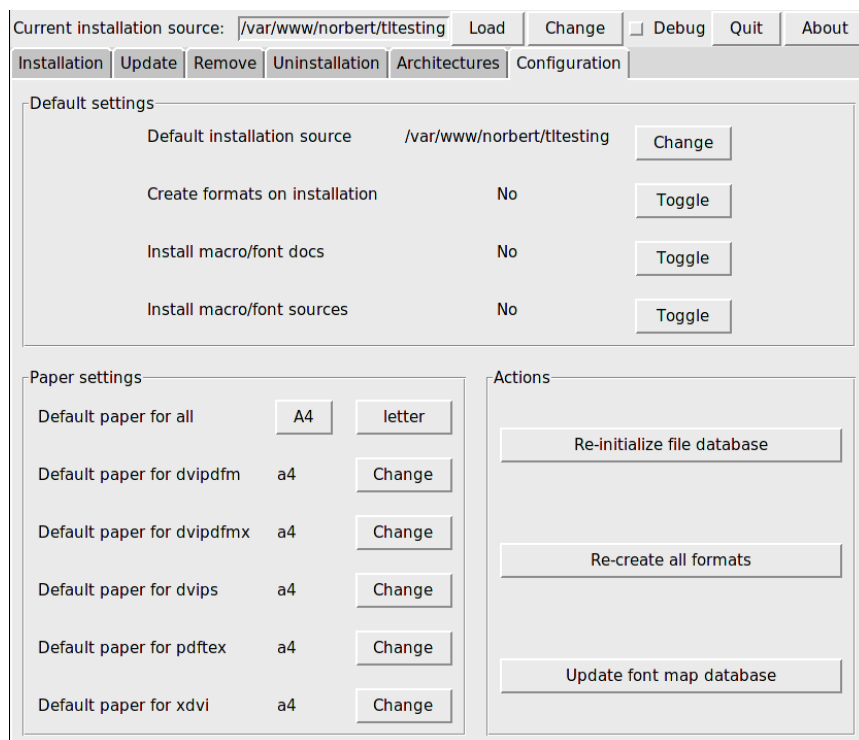


Figure 11. T_EX Live Manager GUI config screen

What else is there?

Besides reworking the whole infrastructure, which is only user-visible in the new installer and the T_EX Live Manager, as with every year, all the programs and packages have been updated. We currently ship around 1400 normal packages, e.g., L^AT_EX and font packages, and around 300 other packages, mostly documentation and a few packages which are T_EX Live internal.

The new player in the game this year is the new engine LuaT_EX (<http://luatex.org>); besides a new level of flexibility in typesetting, this provides an excellent scripting language for use both inside and outside of T_EX documents.

Windows-specific features

To be complete, a T_EX Live installation needs support packages that are not commonly found on a Windows machine. T_EX Live provides the missing pieces:

Perl and Ghostscript. Because of the importance of Perl and Ghostscript, T_EX Live includes ‘hidden’ copies of these programs. T_EX Live programs that need them know where to find them, but they don’t betray their presence through environment variables or registry settings. They aren’t full-scale distributions, and shouldn’t interfere with any system installations of Perl or Ghostscript.

Command-line tools. A number of Windows ports of common Unix command-line programs are installed along with the usual T_EX Live binaries. These include `gzip`, `chktex`, `jpeg2ps`, `unzip`, `wget` and the command-line utilities from the `xpdf` suite. (The `xpdf` viewer itself is not available for Windows, but the Sumatra PDF viewer is based on it: <http://blog.kowalczyk.info/2007/08/software/sumatrapdf/>.)

`fc-cache` helps XeTeX to handle fonts more efficiently.

PS_View. Also installed is PS_View, a new PostScript viewer that is free software; see fig. 12. It also supports viewing of PDF files and is extremely fast. Please contact us with any suggestions, this program is in active development.

dviout This is a DVI previewer which is shipped only in the support directory of the DVD, but you will get it if you use the network update procedure. See fig. 13 for a screenshot.

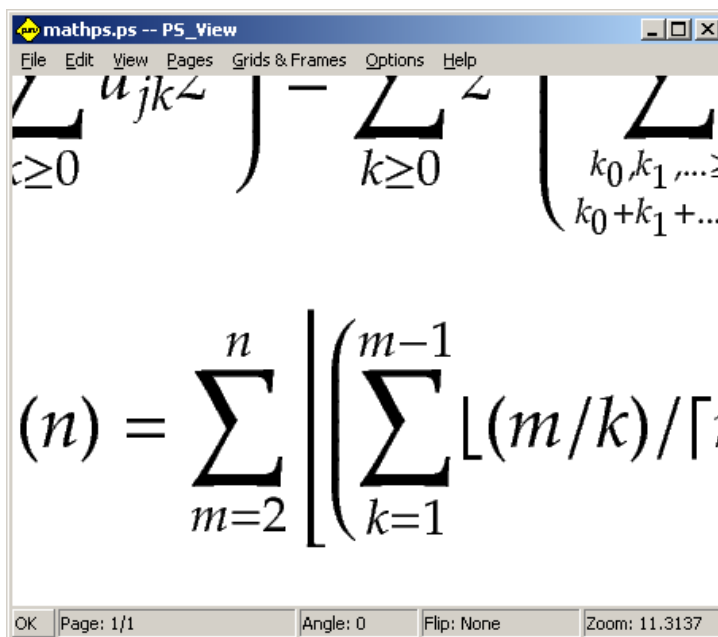


Figure 12. PS_View allows very high magnification, and renders PDF, too

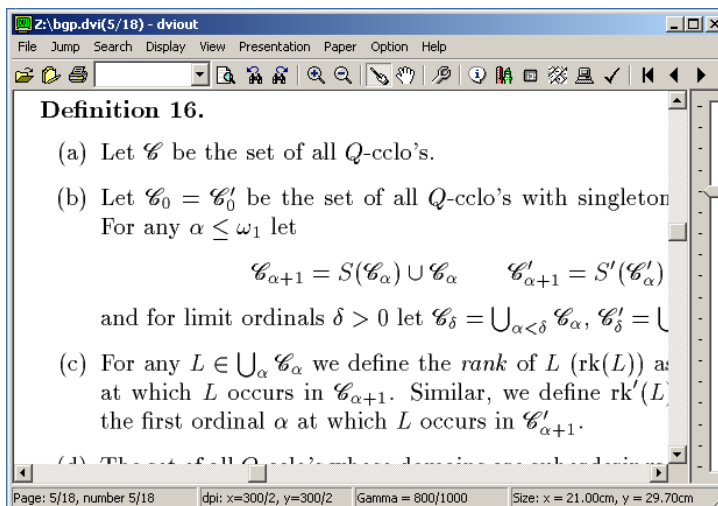


Figure 13. DVIout on Windows

Final remarks and other resources

The T_EX Live Manager is very much work in progress, and its GUI even more. We are adding new functionality frequently, and improving existing functionality to make it more robust. If you see any anomalies don't hesitate to contact us at texlive@tug.org, and we will try to improve it further.

As with most volunteer projects the group of core programmers is quite small. Most of `tlmgr` and its GUI has been programmed by the author with some minor contributions from others. Anyone being more or less able to program Perl is heartily invited to join forces and help us, there are long lists of TODOs for the T_EX Live Manager, let alone for all of T_EX Live.

If you are searching more information on T_EX Live your starting place should be <http://tug.org/texlive/> and the documentation page <http://tug.org/texlive/2008/doc.html>.

The list of people to thank is too long to be included here, please see the online T_EX Live documentation, Chapter 9 (Acknowledgments), for the ever growing list. Of course one name has to be mentioned and that is Karl Berry who with great enthusiasm and perpetual support (and a sometimes critical voice if I was too fast in implementing something!) prepared the T_EX Live 2008 release.

Norbert Preining
preining@logic.at