The teTeX Guide — Installation and Maintenance

Thomas Esser, Dirk Hillbrecht, Craig Bateman

June 1996

Abstract

This documentation describes the main features of the teTeX distribution, a TeX/LaTeX distribution that includes TeX, LaTeX 2ε, METAFONT and many further programs as dvips, xdvi, BibTeX, etc.

teTeX wants to make TeXing from within Unix as easy as possible, and takes full advantage of the efficient and highly customizable Kpathsea-library from Karl Berry.

Contents

1 Introduction

Welcome to the documentation of teTEX, a TEX distribution for UNIX-like platforms. teTEX was mainly set up by Thomas Esser, `te@informatik.uni-hannover.de`, whose work heavily bases on Karl Berry's `Kpathsea`-library, his `web2c`-package and the associated packages as `dvipsk` etc.

Furthermore, there is me. As Thomas had so much trouble with the binaries, the fonts, the directory-organisation, and the additional script files, not to mention the fonts, the directory-organisation, the binaries and of course all of those nifty little additional scripts... Oh, it seems as if I repeat myself... As it is, I offered to write a little documentation and do some beta-testing of his package.

So here we are now. If you find anything about the programs themselves, Thomas is the right address to tell. If you want to make any remarks on the documentation, esp. this file, bother me: Dirk Hillbrecht, `dh@informatik.uni-hannover.de`.

1.1 Features

OK. What is it now, this remarkable new TEX? What makes it so different from other TEXs? Well, in the first place, it was assembled within approximately half a year, thinking about this topic and that feature, testing this and that program, finding one and another bug, having great ideas of a perfect directory tree — and throwing them away again... So you see, this TEX has been done with a lot of attention and passion. We really think it to be superiour to most other TEX implementations.

Furthermore, there are the following highlights:

- Multiplatform usability. teTEX is known to run on the following platforms:
  - Linux on Intel and Alpha platforms;
  - SGI Irix 5.3, 6.2
  - SunOS 4.1.3 on SPARC
  - Solaris 2.4, 2.5 on SPARC
  - FreeBSD 2.1.5 on Intel
  - NetBSD 1.2 on Intel
  - Ultrix 4.3 on Mips
  - IBM RS6000 AIX 3.2.5, 4.1.4, 4.2.0
  - PowerPC AIX 3.2.5, 4.1.4
  - Digital Unix 3.2, 4.0
  - NEXTSTEP (NeXT,Intel,HP-PA,SPARC)
  - Convex C-110, C-210
  - HP-PA/HP-UX 9.xx, 10.xx
  - HP 400/HP-UX 9.xx

Other platforms are planned for the future.

- Fully compile-time independent. teTeX can be installed in any directory in your system. You even do not need to set any environment variables. The binaries will find themselves automatically once they are called. Of course, you can set environment variables to change or enlarge your search paths.
- Installation and maintenance utilities for the whole package make these difficult topics much easier, especially for inexperienced users.
- Very fast file searching using all features of the `Kpathsea`-library v2.6 (even within the web2c package).
- The `kpsetool` as a run time interface to all the search paths of the teTeX programs.
- Ready-to-use font families: cm, dc, charter, oldgerman, ams, pandora, adobe and some special ones.
- Ability to cope with the whole bunch of standard PostScript$^{\mathrm{TM}}$ fonts *even usable with the normal* `xdvi`*-viewer.*
- Latest version of LaTeX 2$_\varepsilon$ (1995/06/01), including the *official* `amstex` package version 1.2 for LaTeX 2$_\varepsilon$ (formerly known as AMS-LaTeX).
- Lots and lots of really useful documentation in `texmf/doc`, even made accessible through WWW-Browsers like Netscape (or Mosaic).

1.2 Series and Locations

The teTeX distribution is split into two series: The *Base Series* and the *Goodies Series*. These series themselves are split up again into several packages (`.tgz`-files) that contain all the system-independent files.

The *Base Series* contains everything that is needed to handle the common TeX-stuff: compile a source, preview the document, convert it into PostScript and print it. If you know how to embed this into your favourite editor and can print PostScript files (e.g. via GhostScript), you will not need more than this series.

The *Goodies Series* brings some useful things to you such as some more dvi-drivers, lots of documentation, and many additional fonts. When having these both series you have everything for a complete TeX-Development-System. Not bad, is it?

Having these two you can do — nothing. As mentioned, these series do not contain any system-dependent files, like e.g. binaries.

But some people want even more! The *Source Series* brings you all the sources needed to compile the teTeX-binaries yourself. Having this one, you may even try to convert teTeX to another platform — as we do. *You will not need the Source Series for normal TeX work.*

1.3 Related software and Requirements

There are some programs not included in teTEX, but which are quite useful indeed:

- X11: Without the graphical environment you will not be able to preview your documents (as long as you do not install an SVGA-based dvi-previewer on your own). Fortunately, X11 is available on most UNIX platforms. The `virtex` and `xdvi` programs need dynamic X11 libraries. As both XFree86-3.1 and XFree-2.X are in common use, these two binaries come in two versions: One is dynamically linked against the X11R6 libraries (from XFree86-3.1) and the other one against the old X11R5 libraries from XFree86-2.X.

- GhostScript: teTEX uses GhostScript for everything concerning the PostScript fonts and for general printing (not using a special dvi driver) on a non-PostScript printer. This program is also part of every larger LinuX-distribution and should be available, too.

Of course you will need some disk space too. These are: approx. 10 MB for the *Base series*, approx. 15 MB for the full *Goodies series* and 13 MB for the full *Source series* (and an additional 10 MB if you actually want to compile them).

In addition, you will need approx. 5 - 20 MB for pk fonts, depending on

- the resolution of your printer

- how much the fonts in your documents vary

- how much you use teTEX

Last not least, your printer(s) must be up and running.

With teTEX you have several possibilities to print your `dvi`-files:

- With `dvips` directly to a PostScript printer. You need to know the logical name of your printer (= the name used by `lpr` or `lp` or whatever your printer access program is) and the METAFONT-mode (the resolution will automatically be calculated during the configuration process).

- With `dvips` to a non-PostScript printer. This requires some configuration of your `/etc/printcap`-file, because a so-called *input filter* needs to convert the PostScript input into an output understood by your printer. For most printers, this can be done with GhostScript (type `gs -help` to get a list of printers supported by your version of GhostScript). The setup of such an input filter (and even much more) can easily be done with the `apsfilter`-program which is available from the German site `ftp-i2.informatik.rwth-aachen.de` in the directory `/pub/Linux/pak/APSfilter` (see the Linux-Software-Map for mirror sites).

- With a specific printer driver such as `dvilj4`. The driver knows what kind of output the printer understands. In general, you only need the logical name of the printer.

1.4 teTEX for emTEX-Users

For many people working with TEX on a PC TEX is equal with emTEX, the best-known and widest spread TEX distribution. Its binaries are OS/2 only indeed[1], so it is not usable under LinuX. But teTEX might be kind of a replacement for it under LinuX: Like emTEX, this should be an easy-to-use-TEX that can be installed by the average user and is fast and convenient. Below, there are the most important changes between emTEX and teTEX:

- teTEX does not need any environment variables to run as long as you keep the given directory structure (but you can use environment variables to make teTEX search your private input directories, e.g. `setenv TEXINPUTS ~/TeX//:` will make TEX search your private macros below the `TeX`-directory in your home directory and then the standard locations).

  You can use the `kpsepath`-program to display the run-time paths.

- Although both teTEX and emTEX are fast (emTEX is only fast when you can use the protected mode binaries on a 386 or higher), it is a major advantage of teTEX to run in a real multitasking environment: you can continue editing your text while the previewer is running and your document is being processed by TEX.

- You can obtain all the sources of teTEX— there are no secrets.

2 Installation

The way you install teTEX depends on what version and source media you have. If you get teTEX as part of a LinuX distribution, just install it the same way you do for other packages of your distribution.

If you get teTEX as a stand-alone package, there are two ways to install: from sources or using precompiled binaries. See the file `INSTALL.src` or `INSTALL.bin`, respectively, for instructions. The binary installation is nothing more than unpacking the archives and running texconfig. The compilation of the sources can be done with a `make world` from the top-level directory. But, you will need recent versions of `gcc`, `flex` and `GNU make` for the compilation.

2.1 `texconfig`

After the installation program has copied all files to their final locations it calls a program called `texconfig` that allows you to configure teTEX to fit your local needs.

3 Using and maintaining teTEX

3.1 The directory structure — Where are my files?

teTEX comes with a very clearly structured directory tree which is described in the following:

[1] It will run under DOS also, but it is not any fun there…

In the first order, we distinguish between site-dependent programs such as `xdvi`, `dvips` or `dvicopy`, which have their files in `bin, man, info` subdirectories of `$TETEXDIR`.

The site independent TEXfiles are installed below the directory `$TEXMF`, which defaults to `$TETEXDIR/texmf`. `texmf` stands for "TEX and more" and is often spoken as "TEX and friends".

Here we find inputs for TEX, METAFONT, BibTEX and all the sources and metrics of the fonts.

The organisation follows the "TeX Directory Standard" (TDS) to a large extent. See file `$TEXMF/doc/help/tds.dvi` for more information about the TDS.

3.2 The input file database

As you perhaps have already noticed, teTEX is fast. Very fast. Quite extraordinarily fast. This is due to a fantastic feature of the `Kpathsea`-library: The *input file database*. This is not much more than the output of a simple `ls -LAR`-command, running over the `$TEXMF`-directory tree. The TEX programs can read this database once at the beginning of their run (which is a fast operation), store it in a hash table and then look up any needed file in this database rather than searching the whole directory tree on the disk (which is — due to the so-called `stat`-operation — very time-consuming).

The database can be found in the `$TEXMF`-directory and is called almost like the command to produce it: `ls-R` and it can be maintained through three different programs:

1. `texhash`. The `texhash` program builds the database from scratch by scanning the whole `texmf`-directory-tree and putting the output of the `ls`-command into the (newly created) `ls-R`-file.

2. `append_db`. This little shell-script is especially for the case that a new font has been created and therefore should be entered into the database. `append_db` gets the font's file name and its location and creates then an `ls -LR`-like entry for this one font at the end of the `ls-R`-file.

What can one do with these scripts now and where are the pitfalls in their use?

For the following, assume a system, where are two users, `adm` and `foo`. Both are non-priviledged, but `adm` installed teTEX and `foo` uses it.

While the installation, `adm` ran `texhash` (automatically during installation). So the `ls-R`-file is his. Now `foo` comes, compiles a file and previews it. Therefore one or more fonts are created and stored — and have to be appended to `ls-R`. This is done by `foo` and therefore with his privileges — and these include assurently not to write into `adm`'s files!

So, we have a problem. There are new files on the disk that are not listed in the database. Each lookup of one of these files will cause time-consuming search operations on the disk, or – if disk searching is disabled (i.e. when the path component is prefixed by !!) – it might happen that those files are not found or that MakeTeXPK or MakeTeXTFM are accidently called.

In the following we give some hints how to avoid such problems:

1. Make `ls-R` world-writeable. This is the easiest solution, and should be safe enough for most sites. teTEX is installed this way when it comes out of the box.

2. Create a new group `tex` and make `ls-R` writable for this group. Then, rename `append_db` to `append_db.sh` and call the renamed shell-script from a small set-GID programm (written in C) with the name `append_db`.

   This will give you some more security (sa ls-R no longer needs to be world-writable), but it requires some more administration whilst the installation (esp. `root`'s privileges to make the `tex`-group), so teTEX does *not* install itself like this in the first place.

In any case, you need to run `texhash` after installing new styles or fonts below `$TEXMF`.

For additional information about the library and file searching in teTEX, see the documentation in the `Kpathsea`-package.

3.3 Adding new fonts

teTEX comes with many fonts included. But perhaps you need more fonts, e.g. your firm specific name font. The question is: Where to put it?

The following table shows where to put the font related files, depending on their type. In general, this type is given by the extension of the filename:

file	to
.fd	$TEXMF/tex/latex/*package*
.mf	$TEXMF/fonts/source/*source*/*typeface*
.tfm	$TEXMF/fonts/tfm/*source*/*typeface*
.afm, .vf, ...	$TEXMF/fonts/{afm,vf,...}/*source*/*typeface*
.pfa, .pfb	$TEXMF/fonts/type1/*source*/*typeface*
.*pk	$TEXMF/fonts/pk/*source*/*typeface*/*mf-mode*

`$TEXMF` is the directory given in the installation process to put the site-independent stuff. To get the run-time value of `$TEXMF` you can run `kpsexpand '$TEXMF'`.

source is the name of the font distributor, such as $\mathcal{AMS}$, Bitstream, Adobe, or `public` for freely distributable fonts. *typeface* is the name of the font family, e.g. Computer Modern `cm`.

3.4 Adding new system-wide styles

A new style should be put below `$TEXMF/tex/`*format*`/`*package* for all formats (e.g. plain, latex, eplain).

To avoid *package* directories containing only a single style file, use `misc` directory for these instead.

Examples:

```
$TEXMF/tex/ams/amssym.tex
$TEXMF/latex/graphics/epsfig.sty
$TEXMF/latex/misc/anysize.sty
```

The documentation for styles is held separately in the directories below `$TEXMF/doc`.

3.5 User-dependent configuration

On a site with lots of users, all these users may have their own style files that are not to be copied into the common teTEX tree. There are several possibilities to make these files visible for the TEX programs:

- The `.sty`-file can be copied into the current directory. This directory is searched by default. Of course, this is only sensible if the style is very proprietary and for just a single TEX source.
- Every user can change the `TEXINPUTS` environment variable. Anyway, they should ensure that teTEX's own paths are searched, too. This can be done by putting a colon in the beginning or at the end of the path or a doubled colon in the middle. Such an empty path component will be replaced by the default path.

  Typical settings for *csh* and *tcsh* could be

  ```
  setenv TEXINPUTS :~/TeX/mystyles
  ```

  or for *bash* or *sh*:

  ```
  export TEXINPUTS=~/TeX/mystyles:
  ```

- At least, the TEX-Administrator himself can take care of the user's style files by putting something like `...:~/TeX//` into the `TEXINPUTS`-statement in the `texmf.cnf` file in teTEX's base directory. The two slashes mean to search recursively in all subdirectories.

3.6 Fonts again: Creation on suspicion, import, export

When you start working with teTEX, you have no `.pk`-files. This is due to the fact that every resolution needs its own `.pk`-files (they contain the pixel information about the letters). So it would not be of any use to you if teTEX came with precalculated fonts for an Epson Stylus (360 dpi) whilst you have a HP Laserjet IV (and need therefore fonts with 600 dpi resolution). teTEX calculates these files "on demand", that means if and only if they are needed.

Nevertheless it is quite annoying to wait for the first documents created with the teTEX-series several minutes, because of METAFONT calculating the needed fonts. Fortunately, there are some possible ways to shorten this time (perhaps down to 0):

1. Calculate a large bunch of fonts just on suspicion. You just start `allcm -m` *mode* or `alldc -m` *mode* — and go away. Depending on the power of your computer these programs will work several minutes, hours or days to calculate all fonts LATEX will need of the Computer Modern or DC-fonts for the given `mode`. On the one hand this might be an overkill, filling lots of your disk space with useless fonts (useless for you, because none of your documents use them), but on the other hand it is sure that there are no fonts missing to show almost any document immediately.
2. Calculate all the fonts needed to show the documents existing on your system. This may be of interest for you if you have a large number of documents and bought

a new printer with a different resolution. Just run the `allneeded` script. This script will search the given directories for `.dvi`-files and run them through `dvips` and (optionally) `dvired` (with the PostScript output sent to `/dev/null :-)`). The nice side-effect of this operation is that the `dvips` will call `MakeTeXPK` to create the needed fonts, and the result of this is, that all the `.dvi`-files on your system are processable without delay — as long as you do not produce new ones that use different fonts, of course `:-)`.

3. Import font files from a different TeX-Installation. This can be done through the program `fontimport`. We even provide a possibility to exchange fonts between teTeX and TeX implementations using short filenames (`dpi300/cmr10.pk` instead of `cmr10.300pk`), just call `fontimport` or `fontexport -d`. For details, refer to `fontimport(1)` and `fontexport(1)`.