The teTEX System: Concepts of Installation, Configuration and Maintenance.

Thomas Esser

Universität Hannover
Institut für Informatik
Lange Laube 22
30159 Hannover, Germany
te@informatik.uni-hannover.de

Abstract. *teTEX is a complete TEX distribution for UNIX platforms that claims to be easy to install, to configure, to maintain and to use. This article describes the underlying basic concepts and design decisions that have been used to achieve this goal.*

1. Introduction

teTEX is a TEX distribution for UNIX systems. It is freely available on the internet [1] in two forms: source code and binary. It is based on the Web2c distribution [3], but contains more programs and a larger directory tree with fonts and macros.

2. General system design

The systems consists of two parts: a platform-specific part containing the programs (one directory for each platform), the UNIX manual-pages and the GNU info-pages, and a platform-independent part that is a directory tree (called "`texmf`-tree") which contains font files, macro packages and configuration files. The `texmf`-tree is organized according to the "TEX Directory Structure" (TDS) standard [2]. The TDS implies the organization of files in many subdirectories, so finding the directory in which a file is located is a performance problem.

This problem is solved by using the Kpathsea[1] library by Karl Berry. In Section 2.3, we explain a solution to that performance problem. All programs that need to access files in the `texmf`-tree use the Kpathsea library directly and all scripts that need to access files there use the Kpathsea library indirectly by calling an interface program named `kpsewhich`. The first use of `kpsewhich` was to find files in the `texmf`-tree (like the UNIX program `which`), but other functions have been implemented and these can be used by calling `kpsewhich` with appropriate arguments.

It is a design decision that all programs and scripts use the Kpathsea library for accessing files in the `texmf`-tree. The following benefits of the Kpathsea library show why this decision was made:

2.1. Dynamic auto-configuration

The programs of the teTEX distribution need to access several kinds of file in the `texmf`-tree, e.g. “TEX font metric files”, “BibTEX style files”, “virtual font files” and so on. The Kpathsea library defines search paths for each of these file types. An auto-configuration feature dynamically adjusts all search paths to the root directory where the distribution is installed. That way, it is not necessary to install the distribution in some special directory and no special privileges are needed for installing and using teTEX.

This auto-configuration is implemented in the following way: when a program is started, it searches for the directory in which it is installed[2] and sets a few environment variables according to the result. `$SELFAUTOLOC` is set to the name of the directory in which the program is found, `$SELFAUTODIR` is set to the name of the parent of that directory and `$SELFAUTOPARENT` is set to the name of the parent directory of `$SELFAUTODIR` (see Figure 1).

Variable	Value
`SELFAUTOLOC`	`/usr/teTeX/bin/ix86-linux`
`SELFAUTODIR`	`/usr/teTeX/bin`
`SELFAUTOPARENT`	`/usr/teTeX`

Figure 1 – SELFAUTO variables for `/usr/teTeX/bin/ix86-linux/latex`.

These SELFAUTO variables are used to find a runtime configuration file (named `texmf.cnf`) and to set up search paths from that file (see Figure 2).

1 'K' stands for 'Karl' and 'sea' stands for 'search'. The current maintainer of the Kpathsea library is Olaf Weber. We cannot fully describe all features of that library in this paper. For a full description, see the Kpathsea manual, available at [3].

2 It searches for itself along `$PATH` and then follows symbolic links.

```
TETEXDIR   = $SELFAUTOPARENT
TEXMFMAIN  = $TETEXDIR/share/texmf
TEXMF      = $TEXMFMAIN
VFFONTS    = $TEXMF/fonts/vf//:.
```

Figure 2 – Search path definitions in `texmf.cnf`.

The main part of the computation of search paths is variable expansion. Let us consider the search path for virtual font files. The `texmf.cnf` specifies a `VFFONTS` variable and its value overrides the compile-time search path. By substituting all intermediate variables, you will see that the value of `$VFFONTS` is equivalent to `$SELFAUTOPARENT/share/texmf/fonts/vf//:.` . Since this search path (as well as all the other search paths in `texmf.cnf`) is relative to one of the `SELFAUTO` variables, it automatically adjusts if you move the complete teTeX installation to another location in your file system. It is important that the name of the root directory where the distribution is installed is not explicitly written into any configuration file and only computed dynamically.

2.2. Unified search path specification

Before Kpathsea implemented common file lookup methods, different programs used different variable names for the same search paths and the syntax and semantics were different.

2.3. Filename database

When searching for a file, Kpathsea tries to avoid time-consuming recursive searches on the disk by doing a lookup in a filename database (`ls-R`-file[3]). A filename database lists all files that exist in the same directory or a subdirectory of the filename database and is typically installed in the `texmf`-directory. The `ls-R`-file contains the subdirectory names directly followed by a colon and the names of the files contained in the subdirectory on the following lines. A fragment of such an `ls-R`-file might look like this:

```
./fonts/source/public/cm:
accent.mf
bigacc.mf
bigdel.mf
```

[3] The name is derived from the UNIX command that is used to build the filename database. It is basically `ls -R`.

```
bigop.mf
calu.mf
cmb10.mf
cmbase.mf
...
./fonts/source/public/cmextra:
cmbxcd10.mf
cmbxti12.mf
cmbxti7.mf
```

Figure 3 – Fragment of an `ls-R`-file.

2.4. Automatic font generation

Automatic font generation is supported for font source files (driver files for METAFONT, e.g. for the ec fonts), TEX font metric files and packed bitmap fonts. If searching for a file of one of these types fails, Kpathsea calls an external program which tries to build that file. If the file can be built successfully, the calling program does not notice that the file did not exist at the time when the search started.

We have just seen four benefits of using the Kpathsea library. These are the most important ones, but the library offers many more interesting features which cannot be described here.

3. Installation

The teTEX distribution can be installed in one of two ways: by using precompiled binaries or by compiling the sources.

3.1. Binary installation

To ease the binary installation, a menu-driven installation script is provided. The script was written in the language of the Bourne shell, since a Bourne shell compatible command interpreter is available on most UNIX systems. The main features of the installation script are that it allows us to set up the destination directories and to select which items are to be installed. Basically, it runs in two parts. In the first part, the installer is presented with a menu where the configuration can be set up. We think that feedback is an important issue for a user interface, so we have designed the installation script to show feedback whenever it is possible: the main menu shows the detected platform, a summary of the selected packages including the required disk space, the directory names for the installation and an overview of a few other options (see Figure 4).

In the second part of a binary installation, the files of the distribution are extracted from the archives into their final location and the system will be initialized by creating some important files. This part runs without intervention.

```
===================> teTeX 0.4 installation procedure <====================

        detected platform: Intel x86/Linux (ELF)

    <S> series:
          binaries:            1 out of 31, disk space required:   3430 kB
          other (summary):    27 out of 27, disk space required:  36928 kB
                                               total disk space:  40358 kB

    <D> directories:
          TETEXDIR = /usr/local/teTeX
          TEXMF    = /usr/local/teTeX/texmf

    <O> options:
          [ ] alternate directory for automatically generated fonts
          [ ] create symlinks in standard directories
          [ ] save space for manpages

    Other commands:
          <I> start installation,  <H> help,  <Q> quit

Enter command:
```

Figure 4 – Installation script: main menu.

3.2. Installation from sources

The source distribution is a collection of several packages which we have put together under a new top-level directory. The advantage of this approach (in contrast to several separate packages) is that all programs can be compiled and installed easily together with only a few commands. By using configure scripts (generated by GNU autoconf), it is not necessary to set up system dependencies manually. An installation from the sources is the preferred choice when precompiled binaries are not available for a platform or when changes to the sources are needed.

A feature common to the binary- and to the source-installation is that the destination directory[4] can be chosen freely. This is possible due to the dynamic auto-configuration performed by the Kpathsea library (see Section 2.1).

[4] i.e. the directory where the distribution will be installed.

The files in the TDS directory tree can be shared between different platforms[5] and the only files needed to run the teTEX system on a new platform are the binaries for that platform. If in a binary installation several platforms are selected for installation, for each platform a new directory holding the corresponding binaries will be created. If you use a teTEX installation with binaries for several platforms installed, you have only to set your environment variable `PATH` appropriately for each platform, e.g. include `/usr/local/teTeX/bin/ix86-linux` on a Linux system (on Intel or compatible processor) and `/usr/local/teTeX/bin/sparc-solaris2.5.1` on a Sparc (running Solaris 2.5.1). For a multi-platform installation from sources, install the `texmf`-tree once and compile and install the programs on each platform you need.

4. Configuration

4.1. Using texconfig

We have written a tool named `texconfig` that can help the user with the most common configuration tasks. That tool can be used in command mode or in interactive mode when called without parameters. For the interactive mode, a cursor based utility is used for user interaction (menus, checkboxes, ...). `texconfig` can be used to set up the hyphenation tables stored in the TEX format files, to set up printers (for `dvips`), to set preferences for automatic font generation, to set up the default resolution for previewing (for `xdvi`) and for a few other things. It manipulates configuration files to store the configuration changes.

We will describe the way `texconfig` works for the example of changing the default METAFONT mode (and corresponding resolution) for `xdvi`, `dvips` and METAFONT (see Figure 5).

In the first step, the script parses the file `modes.mf` and extracts the list of available modes which are then put into a scrollable listbox. Once the administrator has selected the preferred default mode (the resolution is shown as a comment for each mode), the following will happen: the mode and corresponding resolution will be set up in the configuration files for `xdvi` and `dvips`. Then, the mode in the `localfont` definition for METAFONT (file `modes.mf`) will be adjusted and a new base file for METAFONT (`mf.base`) will be dumped and installed.

5 They can even be shared between different TEX implementations.

```
+---------------------------- teTeX setup utility -----------------------------+
|  Choose a default mode for xdvi, dvips and metafont. As a rule               |
|  of thumb, choose the mode most commonly used for printing at your site.     |
| +--------------------------------------------------------------------------+ |
| |           agfafzz   AGFA 400PS (406dpi)                                  | |
| |           agfatfzz  AGFA P3400PS (400dpi)                                | |
| |           amiga     Commodore Amiga (100dpi)                             | |
| |           aps       Autologic APS-Micro5 (723dpi)                        | |
| |           apssixhi  Autologic APS-Micro6 (1016dpi)                       | |
| |           atariezf  Atari ST SLM 804 printer (300dpi)                    | |
| |           atarinf   Atari previewer (95dpi)                              | |
| |           atarins   Atari previewer (96dpi)                              | |
| |           atariotf  Atari ST SM 124 screen (101dpi)                      | |
| |           bitgraph  BBN Bitgraph (118dpi)                                | |
| |           bjtenex   Canon BubbleJet 10ex (360dpi)                        | |
| |           boise     HP 2680A (180dpi)                                    | |
| |           canonbjc  Canon BJC-600 (360dpi)                               | |
| |           canonex   LaserWriter Pro 630 (600dpi)                         | |
| +-----------v(+)-----------------------------------------------------------+ |
|                                                                              |
+------------------------------------------------------------------------------+
|                             <  OK  >      <Cancel>                            |
+------------------------------------------------------------------------------+
```

Figure 5 – `texconfig` sample.

4.2. Customizing search paths

Search paths can be customized in several ways. Since the Kpathsea library reads a configuration file, you can globally change a search path by modifying the corresponding variable in the global configuration file, e.g. set

```
VFFONTS = .:$TEXMF/fonts/vf//:/usr/local/TeX/local/vffonts//
```

This example sets up the search path for virtual font files such that the following directories will be searched: the current directory, the directory `fonts/vf` and all subdirectories in each of the `texmf`-trees listed in `$TEXMF` (see Section 5.2) and finally `/usr/local/TeX/local/vffonts` including all subdirectories.

A slightly different way is to use a local `texmf.cnf` configuration file. Kpathsea always reads all configuration files it can find and by default it searches the current directory first. When a search path is defined in more than one configuration file, definitions in earlier files override those in later files. So, a second way to customize search paths is to create a `texmf.cnf` in the current directory.

A third and very common approach is to use environment variables. Let us consider the example that you want TeX to search the directory `~/TeX/macros` for your macro files. You can append your macro directory to the standard search path and use this extended list to define the TEXINPUTS variable, e.g. the following way (assuming a csh-like syntax for defining environment variables):

```
setenv TEXINPUTS .:$TEXMF/tex/{generic:}//:~/TeX/macros
```

Environment variables take precedence over texmf.cnf variables; this environment variable completely redefines the search path. This approach, however, is depreciated, since a much better solution exists:

```
setenv TEXINPUTS :~/TeX/macros
```

This time, the environment variable has an empty first search component. Kpathsea automatically inserts the search path from the configuration file in that place. So, the two definitions are equivalent as long as the first one correctly reflects the standard search path. The second definition always automatically adjusts if the system search path is changed.

5. Maintenance

5.1. Updating the texmf-tree

To ease the administrator's work, from time to time we provide update packages containing updates to files in the texmf-tree. Using the interface program kpsewhich of the Kpathsea library, it is possible to create these update packages as self-extracting shell scripts which automatically install new files at the right location. The update script can automatically determine the location of the texmf-tree by using the -expand-var parameter for kpsewhich:

```
kpsewhich -expand-var='$TEXMFMAIN'
```

5.2. Maintaining local files

It is a good idea to keep local additions separate from the original distribution. That way you can for example replace the distribution easily without losing any of your local additions. A convenient way to add local additions globally is to set up a new texmf-tree and to make this known to all programs by changing the TEXMF-variable in Kpathsea's configuration file texmf.cnf. To set up several texmf-trees, just list them, separated by colons, and put braces around the list, as in the following example:

```
% The texmf tree of the distribution
TEXMFMAIN  = $SELFAUTOPARENT/share/texmf
% A global texmf tree with local additions:
TEXMFLOCAL = /usr/local/TeX/texmf.local
% Each user can have a texmf tree:
HOMETEXMF  = $HOME/texmf
% The TEXMF variable lists all TEXMF trees:
TEXMF      = {$HOMETEXMF:$TEXMFLOCAL:$TEXMFMAIN}
```

Setting up a new `texmf`-tree in such a way automatically adjusts all search paths and all `texmf`-trees are searched in the order given by the `TEXMF` variable. You have to follow the TDS for your `texmf`-tree, since the default search paths are set up for a TDS-compliant directory-tree, or change those search paths where you do not follow the TDS.

You can also put local additions into the standard `texmf`-tree (although this is not recommended). It is important, however, that you update the filename database using the `mktexlsr` script after you have added files to a `texmf`-tree.

Bibliography

[1] *The teTEX Homepage*, `http://tug.org/teTeX/`

[2] TUG Technical Working Group on a TEX Directory Structure, *A Directory Structure for TEX Files*, `http://tug.org/twg/tds/`

[3] *Web2c and Kpathsea*, `http://tug.org/web2c/`