

# teTeX Manual

Thomas Esser

October 2004

## Abstract

teTeX is a distribution of T<sub>E</sub>X and related programs: pdfT<sub>E</sub>X, e-T<sub>E</sub>X, Omega, Aleph, L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt, METAFONT, METAPOST, dvips, xdvi, dvi<sub>pdfm</sub>, BIBT<sub>E</sub>X, makeindex etc.

teTeX aims to make using and maintaining a T<sub>E</sub>X system as easy as possible. The programs are built around the Web2c distribution by Karl Berry and Olaf Weber. A common part of many programs is kpathsea, a library that provides efficient access by name to files stored hierarchically.

This document describes how to use and set up the teTeX system. It does not attempt to be a comprehensive guide, instead it tries to give an overview about what software and documentation is contained in the distribution.

# Contents

<b>1</b>	<b>The Components of teTeX</b>	<b>3</b>
1.1	Web2c . . . . .	3
1.2	TeX extensions: pdfTeX, e-TeX, Omega, Aleph, encTeX . . . . .	3
1.3	DVI drivers: dvips, dviIj, xdvi, dvipdfm, dvipng . . . . .	4
1.4	Makeindex . . . . .	5
1.5	Texinfo . . . . .	5
1.6	UNIX Scripts / Tools . . . . .	5
<b>2</b>	<b>Concepts and configuration</b>	<b>7</b>
2.1	The TeX Directory Structure (TDS) . . . . .	7
2.2	Overview of the predefined texmf trees . . . . .	8
2.3	The file name database (ls-R) . . . . .	8
2.4	Runtime configuration (texmf.cnf file) . . . . .	9
2.5	Using Postscript type 1 fonts . . . . .	9
2.6	Configuration files maintained by texconfig . . . . .	10
2.7	TCX files . . . . .	11
2.8	Creating PDF files . . . . .	11
<b>3</b>	<b>Release notes for teTeX 2.0</b>	<b>12</b>
<b>4</b>	<b>Release notes for teTeX 3.0</b>	<b>12</b>
4.1	New programs / font support / macro packages . . . . .	12
4.2	Changes to web2c . . . . .	13
4.3	Modifications to the directory structure and file searching . . . . .	13
4.4	Changes to texconfig / updmap / fmtutil . . . . .	15
4.5	Changes to pdfTeX . . . . .	15
4.6	pdfetex: the new default TeX engine . . . . .	16
4.7	Changes to xdvi . . . . .	16
4.8	Improved documentation . . . . .	16
<b>5</b>	<b>Resources</b>	<b>17</b>
5.1	Helpindex file for the documentation tree . . . . .	17
5.2	Internet Newsgroups . . . . .	17
5.3	TeX User Groups . . . . .	17
5.4	Mailing Lists . . . . .	18
5.5	Comprehensive TeX Archive Network (CTAN) . . . . .	18
5.6	The TeX Catalogue . . . . .	18
5.7	Frequently Asked Questions (FAQs) . . . . .	19

# 1 The Components of $\text{teTeX}$

This document cannot describe all the programs which are part of  $\text{teTeX}$  in detail, but it tries to give you an overview. This section describes the packages which form the main components of  $\text{teTeX}$ .

## 1.1 Web2c

**Web2c** is a  $\text{T}_{\text{E}}\text{X}$  implementation, originally for Unix, but also running under Windows, Macintosh, DOS, Amiga, and other operating systems. It includes  $\text{T}_{\text{E}}\text{X}$  itself and the following programs:

- METAFONT: a font compiler intended to produce typefaces of high quality [1].
- METAPOST: a program similar to METAFONT, modified to output Postscript code instead of bitmaps. Documentation for METAPOST is available via the command: `texdoc mpman`.
- BIB $\text{T}_{\text{E}}\text{X}$ : a preprocessor to make bibliographies for  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . For documentation, see `texdoc btxdoc` and Appendix B of [2].
- utilities for converting between different font metric and bitmap formats: `gftopk`, `gftodvi`, `gftype`, `pktozf`, `pktype`, `pltotf`, `tftopl`, `vftovp`, `vptovf`.
- DVI utilities: `dvicopy`, `dvitomp`, `dvitype`.
- other tools: `patgen`, `pooltype`, `tangle`, `weave`, `ctangle`, `cweave`.

The main documentation for **Web2c** is the “Web2c manual” and the “ $\text{kpathsea}$  manual”. These can be accessed via `texdoc web2c` and `texdoc kpathsea` respectively.

## 1.2 $\text{T}_{\text{E}}\text{X}$ extensions: pdf $\text{T}_{\text{E}}\text{X}$ , e- $\text{T}_{\text{E}}\text{X}$ , Omega, Aleph, enc $\text{T}_{\text{E}}\text{X}$

Besides the standard  $\text{T}_{\text{E}}\text{X}$  program, the following extensions to  $\text{T}_{\text{E}}\text{X}$  are included in  $\text{teTeX}$ :

**pdf $\text{T}_{\text{E}}\text{X}$**  This can optionally write Acrobat PDF format instead of DVI. The user manual can be accessed by the command `texdoc pdftex-a`. The  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  hyperref package ( `texdoc manual` ) has an option “pdftex”, which turns on all the program features. In DVI mode, pdf $\text{T}_{\text{E}}\text{X}$  works like the usual  $\text{T}_{\text{E}}\text{X}$  with the exception of the availability of additional typographic features.

In PDF mode, pdf $\text{T}_{\text{E}}\text{X}$  supports graphics inclusion for the following graphic formats:

- PNG (portable network graphics)
- PDF (portable document format),
- JPG (jpeg)

- MPS (metapost output)

It does not support EPS (encapsulated postscript), but if you have recent versions of `ghostscript` (version 5.10 or later) and `perl` (version 5 or later) installed, you can use the tool `epstopdf` to convert EPS graphics into PDF. When including a PDF file, `pdfTeX` is able to access parts of a multi-page file: a complete page or a clipped part of a page. This feature extends `pdfTeX` beyond its typesetting capabilities into a versatile tool for handling arbitrary generated PDF files.

**e-TeX** adds a small but powerful set of new primitives, and an extension for right to left typesetting. In default mode, e-TeX is 100 % compatible with ordinary TeX. See `texdoc etex-man` for details.

**Omega ( $\Omega$ )** Omega works internally with 16-bit Unicode characters; this allows it to work directly with almost all the world's scripts simultaneously. It also supports dynamically loaded “ $\Omega$  Translation Processes” (OTPs), which allow the user to define complex transformations to be performed on arbitrary streams of input. Documentation: `texdoc doc-1.12`.

**Aleph** is a TeX engine that combines the good features of eTeX and Omega. The L<sup>A</sup>T<sub>E</sub>X based format of Aleph is called Lamed.

### 1.3 DVI drivers: `dvips`, `dvilj`, `xdvi`, `dvipdfm`, `dvipng`

For printing and previewing DVI files, you need to use one of the DVI drivers that are available:

**dvips** This driver converts DVI files into Postscript. Postscript is a page description language that many laser printers directly support. With the help of the utility `ghostscript`, it is possible to view Postscript documents on screen and to print Postscript documents on non-Postscript printers. This version of `dvips` supports hypertext and partial font downloading. In this release, the search path for config files have been changed: the current directory is no longer searched by default. Additionally, secure mode has been turned on by default. For details, consult the documentation: `texdoc dvips`.

**dvilj** This is a family of drivers to support HP LaserJet (and compatible) printers: `dvilj`, `dvilj2p`, `dvilj4`, `dvilj4l`, `dvilj6`. These drivers are faster than the `dvips` + `ghostscript` alternative (which can also be used to print DVI files on HP LaserJet printers), but they lack a few features like support for virtual fonts, rotated or scaled graphics, etc.

**xdvi** This is a previewer for DVI files under the X Window System. It has support for Postscript specials through Display Postscript, NeWS and `ghostscript`. Hypertext support was also added for this version of `xdvi`. A link can be followed by clicking with Button-1 or Button-2 (open link in a new window) on it. With

the help of source specials, xdvi can interact with most editors in some way. For details, visit the [project page](#).

**dvipdfm** This program can convert dvi files into pdf format, while supporting a rich set of features (via TeX's `\special` scheme): hyperlinks, bookmarks, thumbnails, image inclusion etc. For details consult the documentation [texdoc dvipdfm](#) and the [web page](#).

**dvipng** is a fast DVI to PNG converter.

## 1.4 Makeindex

**makeindex** is a general purpose hierarchical index generator; it accepts one or more input files (often produced by a text formatter such as TeX or troff), sorts the entries, and produces an output file which can be formatted. The formats of the input and output files are specified in a style file; by default, input is assumed to be an `idx` file, as generated by L<sup>A</sup>T<sub>E</sub>X. Documentation: `texdoc makeindex`

## 1.5 Texinfo

**texinfo** is a documentation system. It produces online or printed output from a single source. It uses TeX to typeset documents for printing (`texdoc texinfo`).

## 1.6 UNIX Scripts / Tools

If you are using teTeX under UNIX, you can use the following scripts. More documentation for a specific tool can either be obtained from its UNIX manual page or by running the program with the option `-help`.

**texdoc** allows you to easily access documentation included with teTeX. You only have to remember the file name of the document that you want to access, without the directory part. If you do not specify a file name extension (such as `.dvi`) `texdoc` will try a few default extensions. After searching the file, `texdoc` starts an appropriate viewer. The command `texdoc -help` gives you a list of available command line options. While `texdoc` is a fast and efficient tool if you exactly know what you are searching for, it is of limited use if you don't know the exact name of the documentation file.

**texdoctk** This perl/tk based tool allows you to browse through the available documentation by some topics and to search for keywords. It uses a simple database file for its configuration (named `texdoctk.dat`), which you can customize and extend easily.

**texconfig** allows you to carry out the most common configuration tasks in teTeX. The program can be used in command mode or in interactive mode. For the interactive mode (which is invoked by calling `texconfig` without arguments), a curses

based utility is used for user interaction (menus, check boxes, ...). The command `texconfig help` shows you a list of available command line options (command mode).

`texconfig` can be used to set up  $\TeX$  format files and their hyphenation patterns, to set up printers (for `dvips`) and for a few other things. It manipulates configuration files to store the configuration changes.

If the use of `texconfig` causes a change in some configuration file, `texconfig` will try to save that file into the `texmf` tree that is specified by the `TEXMFCONFIG` variable. Variable (cached) runtime data (such as format files generated by `fmtutil`, map files generated by `updmap`) will be stored in the `texmf` tree specified by the `TEXMFVAR` variable.

**updmap** Some tools in  $\text{te}\TeX$  support Postscript Type 1 fonts: `dvips`, `pdf $\TeX$` , `xdvi`, `dvipdfm` and `gsftopk`, `ps2pk`. The last two in this list just convert outline fonts into a bitmap format which is helpful for applications that don't directly support outline fonts. In an ideal world, all of these programs would share a common configuration file to set up their outline fonts. The reality is different, however, and each tool needs its own configuration. The `updmap` utility was developed to generate these configuration files from a central repository. Adding font definitions for all supported applications from a map file `foo.map` can be as easy as `updmap --enable Map foo.map`. For details, please consult the manual page of `updmap`.

**dvired** This script can be used to print documents formatted for A4 paper 2-up (i.e. two logical pages to one physical page of paper) by scaling the pages to 70.7% of their original size. `dvired` can just be used in the same way as `dvips` (same command line options).

**dvi2fax** This script converts DVI files into FAX G3 format. It uses `ghostscript` (see: <http://www.cs.wisc.edu/~ghost/>) which is not part of  $\text{te}\TeX$ . The DVI file is first converted to Postscript (204 × 196 dpi or 204 × 98 dpi) and then to FAX G3 using `ghostscript` (`faxg3` device).

**allcm, allec, allneeded**  $\text{te}\TeX$ 's DVI drivers generate missing bitmap fonts on demand (the first time they are needed). If you start with a fresh installation, you don't have any bitmap fonts and the delay caused by font generation might be too annoying for you. In that case, the three scripts `allcm`, `allec` and `allneeded` can help you. `allcm` and `allec` create a few DVI files (using  $\LaTeX$ ) which use lots of fonts at various sizes and run these DVI files through `dvips`. This triggers the generation of the most commonly used Computer Modern (`allcm`) and European Computer Modern (`allec`) fonts respectively. You might already have DVI files and want to generate just the bitmap fonts needed by these documents. This can be done by the `allneeded` script. This script will search a given set of directories for DVI files and run them through `dvips`. All these scripts just trigger font generation.  $\LaTeX$  and DVI files generated by `allcm`/`allec` are removed when

the program terminates. Postscript output that is generated by dvips is sent to /dev/null.

These programs accept the command line option “-r” (must be the first option) to generate files for the magnification 707/1000 which is used by dvired. allneeded passes options which correspond to existing file or directory names to find (for locating DVI files). All other options given to any of these three utilities are passed to dvips. So, by passing -D *NNN* -mfmode *some-mode* or -P *some-printer*, you can generate fonts for a specific resolution (*NNN*) and mode (*some-mode*) or for a specific printer (*printer*).

## 2 Concepts and configuration

### 2.1 The T<sub>E</sub>X Directory Structure (TDS)

teT<sub>E</sub>X’s support tree with fonts, macros, documentation and other files (from now on called the “texmf tree”) follows a certain structure: the T<sub>E</sub>X Directory Structure (TDS). This is a standard developed by a T<sub>E</sub>X Working Group of TUG. The TDS is defined in a way so that can be used by different implementations of T<sub>E</sub>X on different platforms. Today, several T<sub>E</sub>X distributions follow this standard: teT<sub>E</sub>X, T<sub>E</sub>X Live, fpT<sub>E</sub>X and miktex, only to mention some. You need to understand this structure if you want to build your own texmf tree (e.g., with all your local additions) or add files into an existing texmf tree. The list of all texmf trees (optionally using some notation called “brace expansion” and !! modifiers; the kpathsea manual explains this in detail) can be obtained by:

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

Table 1 gives a short overview of the TDS. It shows the proper location inside the TDS tree for several kind of files. The complete documentation for TDS can be accessed by `texdoc tds`. If you want to see some examples, just look at the main texmf tree of teT<sub>E</sub>X. It has several thousand files.

T <sub>E</sub> X macros	tex/⟨format⟩/⟨package⟩/
font files	fonts/⟨type⟩/⟨supplier⟩/⟨typeface⟩/
METAFONT files	metafont/⟨package⟩/
documentation	doc/⟨package⟩/
sources	source/⟨package⟩/
BIBT <sub>E</sub> X files	bibtex/{bst,bib}/⟨package⟩/

Table 1: TDS: an overview

The replaceable parts in this table mean:

⟨*format*⟩ The name of the T<sub>E</sub>X format, e.g., latex or amstex.

⟨*package*⟩ The name of the package to which the file belongs, e.g., babel or seminar.

$\langle type \rangle$  The name of the type of a font file, e.g., `pk` (packed bitmap), `tfm` (tex font metric), `afm` (adobe font metric), `vf` (virtual font), or `source` (METAFONT source).

$\langle supplier \rangle$  The name of the font supplier (to whom the font file belongs), e.g., `adobe` or `urw`.

$\langle typeface \rangle$  The name of the typeface name (for this font file), e.g., `times` or `cm` (for Computer Modern).

It is important to know that the default search paths in  $\text{te}\text{T}\text{E}\text{X}$  rely on this directory structure. So, if you add a file to the wrong directory tree, e.g., a TeX macro somewhere in the `fonts` subtree, that file will not be found correctly.

## 2.2 Overview of the predefined texmf trees

As described in the previous section, each `texmf` tree follows a well defined directory structure. This section lists all predefined `texmf` trees and their intended purpose.

**TEXMFCONFIG** The tree specified by this variable will be used to store configuration data changed by the `texconfig` utility.

**TEXMFVAR** The tree specified by this variable will be used to store (cached) runtime data such as format files, generated map files and bitmap fonts.

**TEXMFMAIN** This variable specifies the `texmf` tree where some vital parts of the system are installed, such as helper scripts (e.g. `web2c/mktexdir`), pool files and other support files.

**TEXMFHOME** The expansion of this variable typically depends on `$HOME`, so this dynamically adjusts for each user to an individual directory. The idea is to allow users to store their own versions of macros, fonts etc. in this tree.

**TEXMFLOCAL** This variable names a directory tree which should be used for system-wide installation of additional or updated macro packages, fonts etc.

**TEXMFDIST** This variable names the directory tree which holds the macros, fonts etc. as originally distributed.

Section 4.4 explains, how to redirect `TEXMFCONFIG` or `TEXMFVAR`.

## 2.3 The file name database (ls-R)

`texmf` trees can get very large and to speed up searching in such a tree, a file name database is used. A file name database exists in the root of each `texmf` tree and has the name `ls-R`. It should list each file in the `texmf` tree. The command `texhash` can be used to build an up-to-date file name database for each `texmf` tree. It should be used after files have been added to a `texmf` tree. However, you don't need to run `texhash` for files added by the automatic font generation or the `texconfig` utility.



## 2.4 Runtime configuration (texmf.cnf file)

Search paths and other definitions (e.g., the static sizes of some arrays in  $\text{\TeX}$  or other programs) can be set up in configuration files named `texmf.cnf`. By changing the definitions in these configuration files (te $\text{\TeX}$ 's main `texmf.cnf` is `web2c/texmf.cnf` in the main `texmf` tree), the behavior of programs can be changed without recompiling them. Chapters 3 and 4 of the `kpathsea` manual ([texdoc kpathsea](#)) describe the path searching configuration in detail. Section 2.5 of the `Web2c` manual ([texdoc web2c](#)) describes some interesting runtime parameters that you might want to change.

Some changes to the array sizes require you to rebuild the dump files that the program uses. Run the command `texconfig init` to rebuild all dump files after you have changed one of the array sizes.

This implementation of  $\text{\TeX}$  can read and write files (as can every implementation of  $\text{\TeX}$ ) and it can also call external commands (via the `\write18` stream). Some variables in the `texmf.cnf` file control access to these features. The possibility to call external commands can be turned on or off (default is off). Access to file beginning `."` is disallowed in restricted mode (default for reading files). In paranoid mode, file access is even more restricted and you cannot access files outside the current directory tree (default for writing files). If the first line of a document starts with `%&`, it can be used to pass the name of the format file and/or a TCX file. Parsing the first line, is an extension that can be turned on or off (default is off) in the `texmf.cnf` file.

## 2.5 Using Postscript type 1 fonts

For every font you use with  $\text{\TeX}$ , a TFM ( $\text{\TeX}$  font metric) file is needed. Type 1 fonts usually do not have the same encoding that is used by  $\text{\TeX}$ , so additional metrics that do some re-encoding (virtual font files) are often needed. For a lot of font families, these font metric files and additional map files that you need (see below) can be found on CTAN servers in the directory `fonts`. If support for your fonts cannot be found there, you can use the `fontinst` utility (documentation: `texdoc fontinst`) to create these.

Postscript type 1 fonts can be used by `dvips`, `dvipdfm`, `xdvi`, `gsftopk`, `ps2pk` and `pdf $\text{\TeX}$` . All of these programs require that you set up map files for these fonts. To ease the process of adding map file entries to the configuration files that are used by these tools, you should follow the following steps:

- if your fonts already come with a map file, put that file into the `fonts/map/dvips/misc` directory in the main `texmf` tree.
- otherwise, you will need to create a map file yourself, using the syntax as described in the `dvips` manual (`texdoc dvips`); make sure to set up these fonts as "download fonts", not as "built in" fonts.
- run the command `texhash` and then make the map file known to `updmap` by running the command `updmap --enable Map file.map` (where `file.map` denotes the filename of your map file)

The programs `gsftopk` and `ps2pk` convert Postscript type 1 fonts into bitmap fonts and make these fonts accessible to DVI drivers that do not directly support Postscript type 1 fonts. This conversion is automatically invoked by the `mktexpk` script. That script calls `gsftopk` by default. If you do not have installed the `ghostscript` program (which `gsftopk` needs), or if you want to use `ps2pk` for other reasons (e.g., because it is usually faster) you just need to define the variable `ps_to_pk` to `ps2pk`. This variable can be set in your environment or in the `mktex.cnf` file (see below).

## 2.6 Configuration files maintained by `texconfig`

The `texconfig` utility is a user interface for changing the configuration of the  $\text{\TeX}$  system. The configuration is stored in several files. This section documents the names of the configuration files, their location in the `texmf` tree and their content. This explains how `texconfig` works and enables you to manually configure parameters which are not supported by `texconfig`.

If the variable `$TEXMFVAR` is set (in a `texmf.cnf` file or in the environment), its value is used as the name for a directory which holds the files that `texconfig` generates. `texconfig` also checks for the existence of each of the configuration files in the `$TEXMFVAR` tree and copies missing files from the main `texmf` tree. When a `$TEXMFVAR` tree is used, only the configuration files in that `texmf` tree are modified; those in the main tree remain unchanged.

- `dvips/config/config.ps` stores configuration information for `dvips`. The default values are: 600 dpi resolution; `ljfour` `METAFONT` mode; A4 paper; offset for printing: 0pt,0pt; output goes to `lpr` command.
- `dvipdfm/config` defines the defaults for `dvipdfm`, e.g. the default paper size, the command to convert encapsulated postscript graphics into PDF etc.
- `tex/generic/config/pdftexconfig.tex` This file sets some defaults for `pdf $\text{\TeX}$` , e.g. the default paper size. After changing this file, you have to rebuild the format files by using the command `fmtutil --all`
- `web2c/updmap.cfg` controls how `updmap` generates map files for all supported tools. Besides some general parameters, all active map files are listed here.
- `xdvi/XDvi` This file sets some defaults for `xdvi`. It is read via the app-default mechanism of X11. You can override these app-defaults as usual (i.e. via a file `~/.Xdefaults` or with resources managed by `xrdb`). The file sets the following defaults: 600 dpi resolution; `ljfour` `METAFONT` mode; A4 paper; initial shrink factor of 8; Netscape as the web browser; “thorough” handling of overstrike characters. The manual page `xdvi(1)` explains all of these and other parameters that can be set.
- `web2c/fmtutil.cnf` This file defines which format files are built (and how) and which file can be used to customize the hyphenation patterns that are loaded into

these formats. The programs `fmtutil` and `texlinks` (which are automatically called if the formats are set up via `texconfig`) operate on this file. `fmtutil` can be used to create the format files according to the “rules” defined in `fmtutil.cnf` (for a brief description, just call `fmtutil --help`). If you define a new format file, you usually also need a symbolic link with the name for the format to the appropriate  $\TeX$  engine (e.g., `latex`  $\rightarrow$  `tex`). To create these links, just call the `texlinks` script.

## 2.7 TCX files

TCX ( $\TeX$  character translation) files help  $\TeX$  support direct input of 8-bit international characters if fonts containing those characters are being used. Specifically, they map an input (keyboard) character code to the internal  $\TeX$  character code (a superset of ASCII).

$\text{\texttt{te}}\TeX$  has the TCX files `il1-t1.tcx` and `il2-t1.tcx` which support ISO Latin 1 and ISO Latin 2, respectively, with Cork-encoded fonts (a.k.a.: the T1 encoding). TCX files for Czech, Polish, and Slovak are also provided.

All TCX files that are distributed as part of  $\text{\texttt{te}}\TeX$  can be found in the `web2c` subdirectory of the main `texmf` tree; their file name extension is `.tcx`.

You can specify a TCX file to be used for a particular  $\TeX$  run by specifying the command-line option `-translate-file=tcxfile` or (preferably) specifying it explicitly in the first line of the main document `%& -translate-file=tcxfile`. Note, however, that parsing the first line of an input file in an extension that is disabled by default and has to be turned on via command line switch (`-parse-first-line`) or in the `texmf.cnf` file (see section 2.4).

When using TCX files, you usually must not use  $\text{\texttt{L}}\TeX$ ’s `inputenc` package. The TCX file `cp8bit.tcx` is an exception to this rule. The map defined in that file is the identity, i.e. the characters are all mapped to their original position. The purpose of this TCX file is not the mapping it describes, but the side-effect that all characters (even those with positions above 127) are considered to be printable and can be written to the terminal and into log files.

## 2.8 Creating PDF files

If you want to create PDF documents with the help of  $\TeX$ , there are at least three different ways to do this

1. translate your  $\TeX$  sources directly into PDF by using `pdf $\TeX$` .
2. translate DVI files generated by  $\TeX$  into PDF by using the `dvipdfm` program (now included in  $\text{\texttt{te}}\TeX$ ).
3. translate a Postscript file generated by  $\TeX$  and `dvips` into PDF by using Adobe Acrobat or the `ps2pdf` utility included in `ghostscript`.

When using `ps2pdf`, you should make sure to use at least version 6.50 of `ghostscript`. Earlier versions are known to have serious restrictions on creating PDF output.

No matter which approach you use, there is one common rule when creating quality PDF files: you should avoid bitmap fonts. They just display very poorly when used in PDF documents.

Using only the fonts provided by  $\text{\TeX}$ , you have more choices for which fonts to use. The following typeface families are included in Postscript type 1 format:

- Computer Modern and the AMS fonts (extended versions with polish, czech and slovak and russian letters are available, too); a special extension to Computer Modern providing lots of additional characters (most, but not exclusively accents) are the Latin Modern fonts
- the full set of the 35 basic “LaserWriter fonts” (see `psnfss` documentation, supporting other fonts are `pazo`, `tx`/`px` fonts)
- Bitstream Charter

The EC fonts are not freely available in type 1 format. But, if you have a  $\text{\LaTeX}$  document that uses EC fonts, you usually have two ways to get around this problem. The first is to stop using EC fonts—which can in most cases be done by switching to the Latin Modern fonts. Usually, the EC fonts are activated by `\usepackage[T1]{fontenc}` or `\usepackage{t1enc}` and you just have to add `\usepackage{lmodern}`. The second is to use the CM-SUPER Type 1 fonts, not included in  $\text{\TeX}$ , but available on CTAN servers in the directory `fonts/ps-type1/cm-super`. This package is pretty huge, but it contains all EC fonts in outline format and much more.

### 3 Release notes for $\text{\TeX}$ 2.0

Some default settings of “`tex`, the Web2C implementation of  $\text{\TeX}$ ” have changed. We determined that some extensions were in fact in conflict with the strict definition of  $\text{\TeX}$  as laid down by Knuth.

The most notable change is that parsing of `%&` constructs in the first line of an input file is now disabled by default – it can be enabled in `texmf.cnf` if you desire this. In that case, `tex` will announce itself as “`TeXk`” and print an additional banner line saying that `%&`-line parsing is enabled.

We may encounter more places where the default behaviour is not what it should be, and proceed to make this optional (and by default off) in future versions of Web2C.

### 4 Release notes for $\text{\TeX}$ 3.0

This section briefly describes what has changed since the last major release.

#### 4.1 New programs / font support / macro packages

Two programs (see section 1) have been added with this release: `Aleph` and `dvipng`. Many macro packages have been added, too. The largest additions are the packages

beamer and memoir. In the fonts sections, some additions and updates have happened, too. The largest change in this area is the addition of the Latin Modern Fonts (`lmodern`) in Postscript Type 1 format. These fonts are not as exhaustive as the `cmsuper` fonts, but they are of very good quality and sufficient for languages which use latin characters. Using the Latin Modern Fonts is in most cases preferable over using the `ae` fonts (e.g. PDF files with accents are searchable, text extraction works better).

## 4.2 Changes to web2c

- $\TeX$  now uses the new `tex.web` (version 3.141592) from December 2002 (fixed `\xleaders`, `glueset`, weird alignments).
- `enc $\TeX$`  (see `texdoc encdoc-e`), a  $\TeX$  extension by Petr Olšák for input reencoding is now available (for non Omega based engines). The new functionality is enabled by setting the `-enc` switch at format generation time. It defines 10 new primitives which can be used to control  $\TeX$ 's internal character translation tables and proper multibyte input (e.g. for handling UTF-8).
- The `\input` primitive in `tex` (and `mf` and `mpost`) now accepts double quotes containing spaces and other special characters. Typical examples:

```
\input "filename with spaces"    % plain
\input{"filename with spaces"}    % latex
```

See the Web2C manual for more: [texdoc web2c](#).

## 4.3 Modifications to the directory structure and file searching

**split of texmf trees** The instruction of installing  $\TeX$  from the sources that I provide have been changed. The unpacked `texmf` tarball (`$prefix/share/texmf-dist`, set up as `$TEXMFDIST` in `texmf.cnf`) is no longer mixed with the files that are installed via “`make install`” from building and installing the program sources. These files (e.g. format files) that are more tied up with the programs end up in the directory `$prefix/share/texmf` (`$TEXMFMAIN`).

So, the content of the `texmf` tarball remains completely unclanged in `$TEXMFDIST` and can easily be replaced with a new version without losing other runtime files that are not provided elsewhere.

**changed location for font map files** Following a change introduced with version 1.1 of the  $\TeX$  Directory Structure (see `texdoc tds`), font map files (`.map`) are now only searched in subdirectories of `fonts/map` in each `texmf` tree. The subdirectories of `fontname`, `dvips` and `pdftex` are no longer searched for these files. The `texmf` trees from this distribution follows this convention, but you might need to rearrange some files if you maintain a local `texmf` tree.

Within the `fonts/map` tree the files are organized by syntax and package. Known map file syntaxes are `dvips` (this is the most common one), `pdftex` (a superset of the `dvips` syntax, e. g. the `psname` field is optional), `dvipdfm` and `vtex`. If some file is stored within the `fonts/map/dvips` subtree, this does not mean that `dvips` is the only program that accesses these files. Other programs which support the same syntax can use these files as well. It's just that the program `dvips` has given its name for this syntax.

The next directory level specifies the package that the map file belongs to. Example: the file `charter.map` of the `psnfss` package follows the syntax of `dvips`, so it is stored in `fonts/map/dvips/psnfss/charter.map`.

If you happen to see that some application cannot find a map file which is stored in a wrong location, you have to move that file to the right location. Try to find out the syntax and the package that the file belongs to. If unsure, you can always choose “unknown”, since the only restriction for the directory tree below `fonts/map` is that all file names are unique. The precise `<syntax>/<package>` subdirectory does not affect searching. Do not forget to update the filename database (`ls-R`) by running the command `mktexlsr` (resp. `texhash` which is the same).

**changed search path for map files** Map files used to be searched along the `$TEXCONFIG` variable (“`dvips config`” format in terms of `kpathsea` internals). This has been changed to the `$TEXFONTPMAPS` variable (“`map`” format). The new location of the font map files is included in the new default setting of `$TEXFONTPMAPS`, but not in the default setting of `$TEXCONFIG`. The result is that an old application that searches font map files along the “`dvips config`” format will not work.

For the “C” API of `kpathsea` this change means that map files should be accessed using `kpse_fontmap_format` instead of `kpse_dvips_config_format`. For scripts that use `kpsewhich`, one has to make sure that `--format=map` is used to search font map files.

**changed location for font encoding files** Together with font map files (see above), the font encoding files have been given a new location, too. The new location is `fonts/enc/<syntax>/<package>`. So, if you happen to see some application to fail finding a font encoding file, just move it to the right location in the `texmf` tree and update the filename database.

**omission of “engine” directories** Within the `texmf` trees, the directory trees associated with the names of  $\text{\TeX}$  engines (e. g. `etex`, `pdftex`, `omega`, `mltex`) are no longer searched for  $\text{\TeX}$  macro packages. This means that the `$TEXINPUTS` search path now lies completely within the `tex` subtree.

If you are using the above mentioned “obsolete” locations for  $\text{\TeX}$  macro packages, you have to move them into the `tex` directory tree.

## 4.4 Changes to texconfig / updmap / fmtutil

Following the new directory layout, it is desired not to modify the directory tree specified by the `$TEXMFDIST` variable any more. This makes it necessary to redirect generated output (e.g. map files created by updmap, updated config files) somewhere else.

A new approach was taken by the implementation in  $\text{\TeX}$ -3.0: two new variables have been introduced to specify where to store this data:

<code>TEXMFCONFIG</code>	configuration data
<code>TEXMFVAR</code>	variable (cached) runtime data

If a  $\text{\TeX}$  installation is shared among several users, each of them can now use  $\text{\TeX}$ 's configuration tools. A simple way to do this is that each user sets `TEXMFCONFIG` and `TEXMFVAR` to some directory that he can write to. Assuming a bourne shell compatible login shell, the following settings in the user's `.profile` could be used:

```
TEXMFCONFIG=$HOME/.texmf-config; export TEXMFCONFIG
TEXMFVAR=$HOME/.texmf-var; export TEXMFVAR
```

After changing that file, the user needs to log in and create these directories:

```
mkdir $TEXMFCONFIG $TEXMFVAR
```

Now, he can use `texconfig`, `fmtutil` and `updmap`. It is possible, that globally installed format files, font map files etc. have become unavailable by the new setting of `TEXMFVAR`. Running the command `texconfig init` should cure this by recreating all cached data files. To save some disk space, it might be useful to run `texconfig formats` and remove all but the actual needed format definitions.

## 4.5 Changes to pdf $\text{\TeX}$

- All parameters previously set through the special configuration file `pdftex.cfg` must now be set through primitives; `pdftex.cfg` is no longer supported. Some settings (e.g. the default papersize) are loaded via `pdftexconfig.tex` into the format files.
- `\pdfmapfile` and `\pdfmapline` provide font map support from within a document.
- Microtypographic font expansion can be used more easily.  
<http://www.ntg.nl/pipermail/ntg-pdftex/2004-May/000504.html>
- See the pdf $\text{\TeX}$  manual for more: [texdoc pdftex-a](#).

## 4.6 pdfetex: the new default T<sub>E</sub>X engine

teT<sub>E</sub>X uses pdfetex for all formats except “good-old” tex. So, if you run latex, the underlying engine will be pdfetex. Some (broken) T<sub>E</sub>X macros assume that pdfT<sub>E</sub>X is running in PDF generation mode if they detect primitives that pdfT<sub>E</sub>X has introduced (e.g. \pdfoutput). This is wrong, since pdfT<sub>E</sub>X can also be used (and is used) to generate DVI output. A reliable way of detecting PDF output mode is implemented in ifpdf.sty which works for plain T<sub>E</sub>X as well as L<sup>A</sup>T<sub>E</sub>X.

## 4.7 Changes to xdvi

- On supported platforms the default toolkit of xdvi is now Motif. The GUIs for both toolkits (Motif and Xaw) have been updated: There is a page list for easier navigation and improved menus. The Motif version now has a toolbar and a ‘Preferences’ dialog for advanced customizations.

User preferences that are set via this dialog, the ‘Options’ menu and other dialogs are now saved in a file ~/.xdvirc. This file overrides other X defaults, but not the command-line options. (The option ‘-q’ and the X resource ‘.noInitFile’ can be used to disable this feature).

- The new GUI elements can be selectively toggled via the option or the X resource ‘-expertmode’, which should be used instead of ‘-expert’/‘-statusline’.
- The functionality of oxdvi has been merged into xdvi; oxdvi no longer exists as a separate binary.
- Xdvi now supports string search and text selection in DVI files (menu ‘Modes → Text Selection’), printing DVI files and exporting them in PDF, PostScript and plain text format. Pages can be marked with Mouse-2 in the page list to select them for printing/saving.
- By default, xdvi will create a backup copy of the DVI file so that viewing and navigating still works when the DVI file is being written by T<sub>E</sub>X.
- Other new features include: Support for color specials and colored hyperlinks (customizable via the ‘linkstyle option and X resource), a ‘-watchfile’ option, a ‘-unique’ option for loading new files into a running instance of xdvi, support for hyperrefs ‘hdvips’ specials, and a history of recently viewed files via the menu ‘File → Open Recent’.

## 4.8 Improved documentation

- The file doc/index.html in the distributed texmf tree provides a much improved overview of the included documentation and also provides a search facility for various T<sub>E</sub>X related information sources.

This file is generated by a PHP script which is included in teT<sub>E</sub>X as well.



- The PHP script `doc/texdoc.php` is able to offer a web interface to the information provided in `texdoctk` databases.

## 5 Resources

This section describes where you can find further (or more up-to-date) material and support in the world of  $\text{\TeX}$ .

### 5.1 Helpindex file for the documentation tree

The file `index.html` in the root of  $\text{te}\text{\TeX}$ 's documentation tree is a guide for the documentation that is included in  $\text{te}\text{\TeX}$ . It is a good point to start when you want to browse through the documentation or search for the solution of a specific problem.

### 5.2 Internet Newsgroups

If you encounter a problem which might not be  $\text{te}\text{\TeX}$  specific, but rather a general problem with  $\text{\TeX}$  or  $\text{\LaTeX}$  (e.g., "How can I format a section heading in a different way?"), you should not raise your question on one of the mailing lists for  $\text{te}\text{\TeX}$ . In the following newsgroups,  $\text{\TeX}$ -related matters are discussed:

**comp.text.tex** General things about  $\text{\TeX}$ .

**news.answers** FAQs (also  $\text{\TeX}$ -related FAQs).

**comp.answers** FAQs (also  $\text{\TeX}$ -related FAQs).

**de.comp.text.tex** General things about  $\text{\TeX}$  (German).

**fr.comp.text.tex** General things about  $\text{\TeX}$  (French).

**comp.fonts** Font matters.

**comp.programming.literate** Literate programming.

### 5.3 $\text{\TeX}$ User Groups

If you enjoy  $\text{\TeX}$ , you can join a  $\text{\TeX}$  user group to get support and software and help the  $\text{\TeX}$  community by your membership. The web site of the  $\text{\TeX}$  User Group (TUG), <http://tug.org/> has the necessary contact information for several  $\text{\TeX}$  user groups.

## 5.4 Mailing Lists

All teTeX mailing lists are hosted on the same server which is managed by Majordomo software. Administrative requests, e.g., to (un)subscribe or to get an archive of a list are handled by the address: `majordomo@dbb.uni-hannover.de`. To get a list of available commands that the Majordomo server understands, just send the message “help” to the server (in the body of a message, not in the header). The lists are:

**tetex** General discussions + bug reports about teTeX. General TeX matters that are not teTeX-specific are not discussed. Especially general questions about TeX should *not* be directed to this list; use a newsgroup instead.

**tetex-announce** This (moderated, low traffic) list is used for important announcements about teTeX, such as new releases or important updates.

**tetex-pretest** This is used to discuss beta versions of teTeX and to report bugs in these versions. Bug reports about official (non-beta) releases should not be sent here, but to the tetex list.

Some of the packages which are contained in teTeX (e.g., Omega and pdfTeX) have special mailing lists or web resources on their own. The web site of TUG, <http://tug.org/> has links to many of them.

## 5.5 Comprehensive TeX Archive Network (CTAN)

To aid the archiving and retrieval of TeX-related files, a TUG (TeX User Group) working group developed the Comprehensive TeX Archive Network (CTAN). Each CTAN site has identical material, and maintains authoritative versions of its material. These collections are extensive; in particular, almost everything mentioned in this article is archived at the CTAN sites, even if its location isn’t explicitly stated.

The CTAN sites are currently [dante.ctan.org](http://dante.ctan.org), [cam.ctan.org](http://cam.ctan.org) and [tug.ctan.org](http://tug.ctan.org). The organization of TeX files on all these sites is identical and starts at `/tex-archive`. To reduce network load, please use the CTAN site or mirror closest to you. A complete and current list of CTAN sites and known mirrors can be obtained by using the `finger` utility on ‘user’ `ctan@cam.ctan.org` (it also works with the other CTAN hosts); it is also available as file `help/ctan/CTAN.sites` in teTeX’s documentation tree.

## 5.6 The TeX Catalogue

This catalogue lists many TeX, L<sup>A</sup>TeX, and related packages and tools. Most are available worldwide online from CTAN, the Comprehensive TeX Archive Network. Links are provided in this catalogue to available sources and documentation. The teTeX documentation tree contains a version of this catalogue in `help/Catalogue`. The most recent online version is available at

<http://texcatalogue.sarovar.org/>

## 5.7 Frequently Asked Questions (FAQs)

Documents which list frequently asked questions and their answers (in short: FAQs) are collections of solutions to many common problems. The documentation tree of  $\text{\TeX}$  contains the  $\text{\TeX}$  FAQ in the directory `tetex` and the UKTUG FAQ in the directory `help/faq/uktug-faq`. The  $\text{\TeX}$  FAQ can be read by the command `texconfig faq`.

## References

- [1] Donald E. Knuth. *The METAFONTbook*. Addison-Wesley, 1984.
- [2] Leslie Lamport.  *$\text{\TeX}$ : A Document Preparation System*. Addison-Wesley, 2nd edition, 1994.