Baskerville

Articles may be submitted via electronic mail to `baskerville@tex.ac.uk`, or on MSDOS-compatible discs, to Sebastian Rahtz, Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, to whom any correspondence concerning *Baskerville* should also be addressed.

This reprint of *Baskerville* is set in Times Roman, with Computer Modern Typewriter for literal text; the source is archived on CTAN in `usergrps/uktug`.

Back issues from the previous 12 months may be ordered from UKTUG for £2 each; earlier issues are archived on CTAN in `usergrps/uktug`.

Please send UKTUG subscriptions, and book or software orders, to Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121 476 2159. Email enquiries about UKTUG to `uktug-enquiries@tex.ac.uk`.

Contents

I Editorial: what's this issue all about?

Your committee has long aimed to ensure that the UK makes a valuable contribution to the world of TeX: for example, it is possible to claim that the UK TeX archive led the world in comprehensive provision, and members of your committee continue to take active part in maintaining the internationally-linked CTAN system of archives.

The idea of maintaining the UK's profile motivated us to produce the 'Frequently Asked Questions' (FAQ) issues that appeared as 'Christmas Editions' of Baskerville for 1994 and 1995. At the time that we decided to prepare the 1994 FAQ, we had noted the great value of the NTG's 4AllTeX which had at the time just been released. We knew then that we could not hope to produce a CD-ROM in the short term, but we agreed at the time that a similar disc, targeted at Unix^TM systems would be a useful thing to have. The rest is history (and is well explained in the documentation of the CD-ROM, that constitutes the only 'article' of this issue of *Baskerville*).

Members will recall that we offered the first version of this CD-ROM to them, for sale. Considering the economics of the matter, and given an assurance that useful software for other platforms (notably Microsoft- and OS/2-based PCs and Macintoshes) could be included on the disc, your committee decided to circulate every member with a copy in lieu of two issues of *Baskerville*.

I would welcome feedback from members on their views of the value of the offering, and what (if anything) we might do to improve it. I continue to maintain the FAQ: would members welcome its republication, perhaps as Baskerville Vol. 7 No. 6—nominally the 1997 Christmas edition?

As I mentioned in a brief 'post-editorial' in Baskerville Vol. 6 No. 4, the extreme lateness of that issue was almost exclusively my fault. As a result, this double issue is following very closely on the heels of the last, and Vol. 7 No. 1 is also in an advanced stage of preparation. We maintain the fond hope that we will have brought ourselves up-to-date before the end of 1997, but … we continue to rely, to a large extent, on external sources of articles for *Baskerville*. Please do submit them to Sebastian Rahtz, as mentioned in the banner line of each issue of *Baskerville*!

II Mailings for members

The committee has been asked, by a member, for a mailing list of members of the group. We find, on consulting the Data Protection Registrar, that we may not legally provide such a list, and we are investigating what is needed so that we should be able to supply such lists in future. It seems clear that members must be offered an opt-out.

In the interim, we are offering members of the group the opportunity of promoting occasional mailings. We have offered such facilities, on an *ad hoc* basis, in the past: we have carried fliers for books, and on one occasion for the *Scientific Word* system. Our mailing mechanisms are not sophisticated, and significant insertions may attract an extra charge, as will an insertion that increases the mail cost.

Members who wish to take advantage of this service should contact the membership secretary, in the first instance.

III The TeX Live Guide, version 2

Sebastian Rahtz

s.rahtz@elsevier.co.uk

Michel Goossens

m.goossens@cern.ch

1   Introduction

This documentation describes the main features of the **TeX Live** CD-ROM, a TeX/LaTeX distribution for Unix, Windows32, Amiga and NeXT systems, that includes TeX, LaTeX 2ε, METAFONT, MetaPost, many other programs such as Makeindex, dvips, xdvi and BibTeX; and a very complete set of macros, fonts and documentation conforming to the *TeX Directory Standard* which can be used with nearly every TeX setup.

This TeX package uses the Web2c implementation of the programs, which tries to make TeXing as easy as possible, and takes full advantage of the efficient and highly customizable Kpathsea library from Karl Berry. It can be run either directly from the CD-ROM, or installed on a hard disk.

The **TeX Live** runnable systems contain two experimental extensions to normal TeX:

1. εTeX, which adds a small but powerful set of new primitives, and the TeX–XeT extensions for left to right typesetting; in default mode, εTeX is 100% compatible with ordinary TeX. See `share/texmf/doc/html/e-tex/etex.htm` on the CD-ROM for details.
2. pdfTeX, which can optionally write Acrobat PDF format instead of `dvi`; there is no formal documentation for this yet, but the file `share/texmf/tex/pdftex/example.tex` shows how it is used. The LaTeX `hyperref` package has an option 'pdftex' which turns on all the program features.

While εTeX is stable, pdfTeX is under continual development; the version on the CD-ROM may not be stable. Most platforms have version 0.11 of May 7th, but some have a slightly earlier one of May 5th, which may have problems including PNG files.

The entire GUTenberg distribution for Windows is included on the CD-ROM, ready to install, as are the following complete packages:

- OzTeX 3.0 for Macintosh
- CMacTeX 2.6 for Macintosh
- Macintosh utilities (Alpha, Excalibur, etc.)
- MikTeX for Windows 95
- emTeX for MSDOS and OS/2
- TeX shells for Windows and DOS (Winedt, e4t, TeXtelmExtel, emTeXgi)

These are provided unchanged from , and have not been integrated in any way with the rest of the CD-ROM.

1.1   History and acknowledgements

This CD-ROM distribution is a joint effort by the TeX Users Group, the UK TeX Users Group, and the French TeX Users (GUTenberg), with the support of the Dutch, German and Czech/Slovak user groups. Discussion began in late 1993 when the Dutch TeX Users Group was starting work on its 4AllTeX CD-ROM for MSDOS users, and it was hoped at that time to issue a single, rational, CD-ROM for all systems. This was far too ambitious a target, but it did spawn not only the very successful 4AllTeX CD-ROM, but also the TUG Technical Council working group on a *TeX Directory Structure*, which specified how to create consistent and manageable collections of TeX support files. The final draft of the TDS was published in the December 1995 issue of *TUGboat*, and it was clear from an early stage that one desirable product would be a model structure on CD-ROM. The CD-ROM you now have is a very direct result of the working group's deliberations. It was also clear that the success of the 4AllTeX CD-ROM meant that Unix users would benefit from a similarly easy system, and this is the other main strand of **TeX Live**.

We undertook to make a new Unix-based TDS CD-ROM in the autumn of 1995, and quickly identified Thomas Esser's teTeX as the ideal setup, as it already had multi-platform support and was built with portability across file

systems in mind. Thomas agreed to help, and work began seriously at the start of 1996. The first edition was released in May 1996. At the start of 1997, Karl Berry completed a major new release of his Web2c package, which included nearly all the features which Thomas Esser had added in teTEX, and we decided to base the 2nd edition of the CD-ROM on the standard Web2c, with the addition of teTEX's `texconfig` script.

We are particularly grateful to: Karl Berry for extra advice, encouragement, and (of course) for providing the Web2c distribution; Thomas Esser, without whose marvellous teTEX package this CD-ROM would certainly not exist, and whose continual help makes it a better product; and Ulrik Vieth, for checking many assumptions at the start, and providing a great deal of extra material for the documentation tree.

Fabrice Popineau did the excellent port of Web2c 7.0 to Windows 95/NT and provided much help; Andreas Scherer contributed the Amiga compilation; Gregor Hoffleit contributed the TeXview material for NextStep users, and the NextStep binaries. At Florida State University Supercomputer Research Institute, Mimi Burbank arranged access to a slew of different computers to compile TEX on, and acted as an essential guinea-pig whenever asked. Michel Goossens provided access to computers at CERN, and Robin Fairbairns stepped in to provide an Alpha running Linux at Cambridge.

Some of this documentation is drawn from the teTEX guide by Thomas Esser and Dirk Hillbrecht; the catalogue of packages depends very much on the ongoing work of Graham Williams (mailto:Graham.Williams@cbr.dit.csiro.au), who kindly agreed to allow us to use it here. Mimi Burbank, Robin Fairbairns and Ulrik Vieth worked hard to improve this text.

1.2 Future versions

This CD-ROM is not a perfect product! We plan to re-issue it once a year, and would like to provide more help material, more utilities, more installation programs, and (of course) an ever-improved and checked tree of macros and fonts. This work is all done by hard-pressed volunteers in their limited spare time, and a great deal remains to be done. If you can help, don't hesitate to put your name forward!

Corrections, suggestions and additions for future revisions should be sent to:

Sebastian Rahtz
7 Stratfield Road
Oxford OX2 7BG
United Kingdom
mailto:s.rahtz@elsevier.co.uk

Updates, notes, and suggestions will be made available on in `info/texlive`. A WWW page for information and ordering details is at http://www.tug.org/tex-live.html.

2 Structure and contents of the CD-ROM

The CD-ROM top level directories are:

bin The TEX family programs, arranged in separate platform directories;
info Documentation in GNU 'info' format for the TEX system;
macintosh The OzTEX and CMacTeX packages ready to install, plus some other utilities;
man Documentation in Unix man pages for the TEX system;
msdos DOS TEX packages—emTeX, and three TEX shells;
support The source of all programs, including the main Web2c TEX and METAFONT distribution; this directory also includes various bits of TEX-related software which are *not* installed by default, such as MusixTEX support programs, and a complete set of Ghostscript;
share The main support tree of macros, fonts and documentation;
wingut The GUTenberg distribution for Windows; this consists of compressed archives which must be unpacked and installed on a hard disk. Please see the detailed instructions in French;
win32 TEX packages for Windows 95 and NT users (MikTeX, and the original package of the Win32 port of Web2c).

There are also two installation scripts for Unix systems, `install-cd.sh` and `install-pkg.sh`; we discuss them on p. 6.

2.1 The TDS tree

The **TEX Live** `share/texmf` tree consists of various 'collections', each of which has a set of 'packages', of which there are over 400 on the CD-ROM. Normal installation allows the user to copy all of a collection to a local hard disk

from the CD-ROM, but it is also possible to install just one package of a collection. Each of the collections is divided into *basic* (1), *recommended* (2) and *other* (3). The collections are:

ams   The American Mathematical Society macro packages and fonts

bibtex   BIBTEX styles and databases

doc   General guides and documentation in various formats, including HTML and PDF

dvips   Support for Rokicki's dvi to PostScript driver

fonts   Font sources, metrics, PostScript and bitmap forms

formats   Eplain, RevTEX, physxx, texsis, alatex, text1, lollipop, etc.

generic   Extra macros for use with any format

graphics   Macro packages for graphics

lang   Support for non-English languages

latex   LATEX, including official tools and all LATEX 2_ε contributed packages

metapost   Support for MetaPost

plain   Macros for plain TEX

systems   Binaries for Unix platforms

texlive   Basic material for the distribution

The appendix starting on p. 27 lists all the packages in alphabetical order with the collection they are found in, and a brief description. Thus all packages in collection `latex1` are what one must have to get started with LATEX, packages in `latex2` are recommended for most users, and `latex3` contains optional packages. The directory `share/texmf/lists` contains lists of all files in each package (used by the installation package).

3   Installation and use under Unix

You can use the **TEX Live** CD-ROM in three ways:

1. You can mount the CD-ROM on your file system, adjust your `PATH`, and run everything off the CD-ROM; this takes very little disk space, and gives you immediate access to everything on the CD-ROM; although the performance will not be optimal, it is perfectly acceptable on, for instance, PCs running Linux;
2. You can install all or part of the system to your local hard disk; this is the best method for many people, if they have enough disk space to spare (a minimum of about 10 megabytes, or 100 megabytes for a recommended good-sized system);
3. You can install selected packages to work either with your existing TEX system or a **TEX Live** system you installed earlier.

Each of these methods is described in more detail in the following sections.

> **Warning:** This CD-ROM is in ISO 9660 (High Sierra) format, with Rock Ridge extensions. In order to take full advantage of the CD-ROM on a Unix system, your system needs to be able to use the Rock Ridge extensions. Please consult the documentation for your *mount* command to see if it is possible. If you have several different machines on a local network, see if you can mount the CD-ROM on one which *does* support Rock Ridge, and use it from the others.
>
> Linux, FreeBSD, Sun, SGI and DEC Alpha systems should be able to use the CD-ROM with no problems. We would appreciate receiving detailed advice from other system users who also succeed, for future versions of this documentation.
>
> The discussion below about installation assumes you have been able to mount the CD-ROM with full Rock Ridge compatibility.

3.1 Running TEX Live *from the CD-ROM*

The organisation of Web2c means that you can run programs simply by adding the appropriate directory under `bin` on the CD-ROM to your `PATH`, and the support files will all be found with no further ado. The following table shows the list of available directories and the systems they apply to.

alpha-linux	DEC Alpha Linux
alpha-osf3.2	DEC Alpha OS 3.2
amiweb2c	Amiga
hppa11-hpux9.05	HP9000 HPUX 9.05
hppa11-hpux10.20	HP9000 HPUX 10.20
i386-linux	Intel PC with Linux (ELF)
i586-freebsd2.2	Intel PC with Free BSD
i686-linux	Intel Pentium Pro with Linux
mab-nextstep3	NextStep 3
mips-irix4.0.5	SGI IRIX 4.0.5
mips-irix5.3	SGI IRIX 5.3
mips-irix6.3	SGI IRIX 6.3
mips-ultrix4.4	DECstation Ultrix 4.4
rs6000-aix3.2.5	IBM RS 6000 AIX 3.2.5
rs6000-aix4.1.1	IBM RS 6000 AIX 4.1.1
sparc-sunos4.1.3	Sun Sparc Sunos 4.1.3
sparc-solaris2.5	Sun Sparc Solaris 2.5
sparc-solaris2.4	Sun Sparc Solaris 2.4
sparc-linux	Sun Sparc Linux
win32	Windows 95 or NT

You may worry that when you subsequently make fonts or change configuration, things will go wrong because you cannot change files on the CD-ROM. However, you can maintain a parallel, writeable, TEX tree on your hard disk; this is searched before the main tree on the CD-ROM. The default location is `/usr/local/texmf`, but you can override this by setting the `TEXMFLOCAL` environment variable.

Thus *sh* or *bash* users on an Intel PC running Linux who mount the **TEX Live** CD-ROM on `/cdrom` by issuing the command:

```
mount -t iso9660 /dev/cdrom /cdrom
```

might add the following to their `.profile` script:

```
PATH=/cdrom/bin/i386-linux:$PATH
export PATH
```

If in doubt, ask your local system support guru to help you work out how to mount your CD-ROM or which directory to use for your system.

Appropriate support files will be installed on your hard disk the first time you need them. It is a good idea to immediately run the `texconfig` script to initialize things, and check it all works.

3.2 Installing TEX Live *to a hard disk*

All of the necessary steps to install all or part of the distribution on your hard disk are achieved by mounting the CD-ROM, changing to the top-level directory, and typing:

```
sh install-cd.sh
```

(On some Unix systems, you may need to use `sh5` or `bsh`). This works by accessing lists of collections and packages from the CD-ROM, and trying to guess what sort of computer system you are on; it should start by displaying the following:

```
Initializing collections... Done.
Counting selected collections... Done.
Calculating disk space requirements for
                collections...Done.
Initializing system packages... Done.
```

It will then show the main control screen (Figure 1), which lets you change four things:

1. the type of system you are on, or want to install for;
2. the collections you want to install, at *basic*, *recommended* or *other* level;
3. the location on your hard disk to put the files;
4. some runtime behaviour features.

You choose options by typing a letter or number and pressing return. In the example, a Linux ELF system has been detected, the default of all collections to recommended level has been chosen, and the default installation directory is `/usr/local`; note that the disk space required for the current installation configuration is also displayed. If you make a suggested setup, you need about 100 megabytes of disk free; however, the basic setup will only take about 10 megabytes, and you can enhance it with selected packages as you need them.

Under the directory you choose for installation, the installation script will put the binaries in a subdirectory of `bin`, and the support tree in `share/texmf`.

The `options` item lets you decide whether to make new fonts be created in another location (if you want the main package mounted read-only for most users), and whether to make symbolic links for the *man* and GNU *info* pages in the 'standard' locations.

When you choose `<C>` for collections, you will see the display of available collections, the level of installation selected, and the disk space required (Figure 2). You can set alternative levels of installation for each collection, ranging from *none* to *all*. You can either set this for all collections at once, or choose a particular collection and set its level (Figure 3).

When you are finished, return to the main screen, and ask the installation to start. It will take each of the collections and systems that you requested, consult the list of files on the CD-ROM, and build a master list of files to transfer. These will then be copied to your hard disk, and the initialization sequence run (creating format files etc.). When this has finished, all you need do is add the correct subdirectory of `bin` in the TeX installation to your path, and start using TeX. If you want to move the binaries up one level, e.g. from `/usr/local/bin/alpha-osf3.2` to `/usr/local/bin`, you need to edit `share/texmf/web2c/texmf.cnf` and change the line

```
prefix = $SELFAUTOPARENT
```

to

```
prefix = $SELFAUTODIR
```

You can of course change the value of `prefix` to any directory you like, and move the support directory there.

3.3 Installing individual packages from TeX Live to a hard disk

You may want to use the **TeX Live** CD-ROM to either update an existing setup, or add features to an earlier installation from the CD-ROM. The main installation program is intended for the first time only, and subsequently you should use the `install-pkg.sh` script on the CD-ROM. Run this by mounting the CD-ROM, changing to the mounted directory, and typing

```
sh install-pkg.sh options
```

The script supports nine options; the first four let you set the individual package you want to install, the whole collection (i.e., `ams2`), the name of the mounted CD-ROM directory, and the name of the directory containing the list files (normally these latter two will be set automatically):

    --package=*name*

    --collection=*name*

    --cddir=*name*

    --listdir=*name*

What actually happens is controlled by four more switches; the first two allow you to exclude documentation or source files from the installation; the third stops the default action of running `MakeTeXls-R` on completion to rebuild the file database, and the last does nothing but list the files that would be installed:

    --nodoc

    --nosrc

    --nohash

    --listonly

Finally, you can specify that instead of installing the files, the script should make a *tar* archive in a specified location:

    --archive=*name*

```
==================> TeX Live installation procedure <==========
===> Note: Letters/digits in brackets indicate menu items <===
===>         for commands or configurable options          <===
    Detected system: Intel PC with Linux (ELF)
    <C> collections:    21 out of 30, disk space required: 163955 kB
    <S> systems:         1 out of 20, disk space required:   7946 kB
                                total disk space required: 171901 kB
    <D> directories:
          TEXDIR = /usr/local
    <O> options:
          [ ] alternate directory for automatically generated fonts
          [ ] create symlinks in standard directories
    Other commands:
          <I> start installation,  <H> help,  <Q> quit
Enter command:
```

Figure 1. Installation screen, example 1

```
Current collections setup:           total size : 171901 kB
==============================================================
        name      selection         size
  <1>   ams       [recommended]     6359 kB
  <2>   bibtex    [recommended]     6584 kB
  <3>   doc       [recommended]    26531 kB
  <4>   dvips     [recommended]      563 kB
  <5>   fonts     [recommended]    21862 kB
  <6>   formats   [recommended]     1003 kB
  <7>   generic   [recommended]      501 kB
  <8>   graphics  [recommended]    10373 kB
  <9>   lang      [recommended]     3287 kB
  <W>   metapost  [recommended]     1280 kB
  <X>   latex     [recommended]    28333 kB
  <Y>   plain     [recommended]      756 kB
  <Z>   texlive   [recommended]    56523 kB
                        SUM:    163955 kB
==============================================================
  global commands: select <n>one / <b>asic / r<e>commended / <a>ll
                for all collections
  <R>   return to platform menu
  <Q>   quit
Enter command to modify current selection:
```

Figure 2. Installation screen, example 2

```
Collection: Fonts
================================================================================
Fonts, including metrics, virtual fonts and sources
================================================================================
  <N>  no packages
  <B>  basic packages              [  2007 kB]
  <E>  basic + recommended packages  [ 21862 kB]
  <A>  all packages                [ 34303 kB]
================================================================================
  <R>   return to collection menu
  <Q>   quit
Enter command:
```

Figure 3. Installation screen, example 3

Thus, if we simply wanted to see the files that make up the package `fancyhdr` before we installed, our command and output would be as follows:

```
sh install-pkg.sh --package=fancyhdr --listonly
texmf/doc/latex/fancyhdr/fancyhdr.dvi
texmf/doc/latex/fancyhdr/fancyhdr.tex
texmf/lists/latex3/fancyhdr
texmf/source/latex/fancyhdr/README
texmf/source/latex/fancyhdr/fancyheadings.new
texmf/tex/latex/fancyhdr/extramarks.sty
texmf/tex/latex/fancyhdr/fancyhdr.sty
texmf/tex/latex/fancyhdr/fixmarks.sty
```

Other examples of usage are:

- Install the LATEX package `arseneau`:

  ```
  install-pkg.sh --package=arseneau
  ```

- Install the LATEX package `alg` with no source files and no documentation:

  ```
  install-pkg.sh \
        --package=alg --nosrc --nodoc
  ```

- Install all the packages available in the 'extra' Plain TEX collection:

  ```
  install-pkg.sh --collection=plain3
  ```

- Place all files which are need for PSTricks in a tar file in `/tmp`:

  ```
  install-pkg.sh --package=pstricks \
          --archive=/tmp/pstricks.tar
  ```

3.4 `texconfig`

After the installation program has copied all files to their final locations, you can call a program called `texconfig` that allows you to configure the system to fit your local needs. This can be called at any other time to change your setup, with a full-screen (which requires the `dialog` program) or command-line interface. It should be used for all maintenance, like changes of installed printers, or rebuilding of the file database. Both modes have help text to guide you through the facilities.

3.5 Building on a new platform

If you have a platform for which we have not provided binary sources, you will need to compile TEX and friends from scratch. This is not as hard as it sounds. What you need is all in the directory `support/texk-7.0` on the CD-ROM.

To compile TEX, you should get gcc, flex and a recent version of GNU make. gcc-2.5.8, flex-2.4.7 and GNU make-3.72.1 or newer should be fine. You may be able to work with other C compilers and Make programs, but you will need a good understanding of building Unix programs to sort out problems.

You should first install the support tree from the **TEX Live** CD-ROM (do a basic install, with no system binaries chosen). Then copy the `texk-7.0` directory to your disk, and run

```
 configure -prefix=$TEXMF
```

where $TEXMF is the place where you installed **TEX Live**.

Now type `make install-exec` and relax...

4   A user's guide to the Web2c system

Web2c contains a set of TEX-related programs, i.e., TEX itself, METAFONT, MetaPost, BIBTEX, etc. The original implementation was by Tomas Rokicki, who in 1987 developed a first TEX-to-C system adapting change files under Unix, which were primarily the work of Howard Trickey and Pavel Curtis. Tim Morgan became the maintainer of the system, and during this period the name changed to Web-to-C. In 1990, Karl Berry took over the work, assisted by dozens of additional contributors. The latest result is Web2c Version 7, which was released in February 1997, and forms the basis of the present **TEX Live** CD-ROM.

The Web2c 7.0 system runs on Unix, Windows 95/NT, DOS, Amiga, and other operating systems. It uses Knuth's original sources for TEX and other basic programs written in `web` and translates them into C source code. Moreover, the system offers a large set of macros and functions developed to augment the original TEX software. The most commonly used components are:

bibtex Maintaining bibliographies.

dmp troff to MPX (MetaPost pictures).
dvicopy Virtual font expansion.
dvitomp DVI to MPX (MetaPost pictures).
dvitype DVI to human-readable text.
gftodvi Generic font proofsheets.
gftopk Generic to packed fonts.
gftype GF to human-readable text.
makempx MetaPost label typesetting.
mf Creating typeface families.
mft Prettyprinting METAFONT source.
mpost Creating technical diagrams.
mpto MetaPost label extraction.
newer Compare modification times.
patgen Creating hyphenation patterns.
pktogf Packed to generic fonts.
pktype PK to human-readable text.
pltotf Property list to TFM.
pooltype Display WEB pool files.
tangle WEB to Pascal.
tex Typesetting.
tftopl TFM to property list.
vftovp Virtual font to virtual property list
vptovf Virtual property list to virtual font.
weave WEB to TEX.

The precise functions and syntax of these programs are described in the documentation of the individual packages or of Web2c itself. However, knowing a few principles governing the whole family of programs will help you to benefit optimally from your Web2c installation.

All programs honor the standard GNU options:

-help print basic usage summary.
-verbose print detailed progress report.
-version print version information, then exit.

For locating files the Web2c programs use the path searching library Kpathsea. This library uses a combination of environment variables and a few configuration files to optimize searching the TEX directory tree. Web2c 7.0 can handle more than one directory tree simultaneously, which is useful if one wants to maintain TEX's standard distribution and local extensions in two distinct trees. To speed up file searches the root of each tree has a file ls-R, containing an entry showing the name and relative pathname for all files "hanging" under that root.

4.1 *Kpathsea path searching*

Let us first describe the generic path searching mechanism of the Kpathsea library.

We call a *search path* a colon- or semicolon-separated list of *path elements*, which are basically directory names. A search path can come from (a combination of) many sources. To look up a file "my_file" along a path ".:/dir", Kpathsea checks each element of the path in turn: first ./my_file, then /dir/my_file, returning the first match (or possibly all matches).

In order to adapt optimally to all operating systems' conventions, on non-Unix systems Kpathsea can use filename separators different from "colon" (":") and "slash" ("/").

To check a particular path element p, Kpathsea first checks if a prebuilt database (see "Filename database" on p. 14) applies to p, i.e., if the database is in a directory that is a prefix of p. If so, the path specification is matched against the contents of the database.

If the database does not exist, or does not apply to this path element, or contains no matches, the filesystem is searched (if this was not forbidden by a specification starting with "!!" and if the file being searched for must exist). Kpathsea constructs the list of directories that correspond to this path element, and then checks in each for the file being searched for.

The "file must exist" condition comes into play with VF files and input files read by TEX's \openin command.

–11–

Such files may not exist (e.g., `cmr10.vf`), and so it would be wrong to search the disk for them. Therefore, if you fail to update `ls-R` when you install a new VF file, it will never be found.

Each path element is checked in turn: first the database, then the disk. If a match is found, the search stops and the result is returned.

Although the simplest and most common path element is a directory name, Kpathsea supports additional features in search paths: layered default values, environment variable names, config file values, users' home directories, and recursive subdirectory searching. Thus, we say that Kpathsea *expands* a path element, meaning transforming all the specifications into basic directory name or names. This is described in the following sections in the same order as it takes place.

Note that if the filename being searched for is absolute or explicitly relative, i.e., starts with "/" or "./" or "../", Kpathsea simply checks if that file exists.

Path sources

A search path can come from many sources. In the order in which Kpathsea uses them:

1. A user-set environment variable, for instance, `TEXINPUTS`. Environment variables with a period and a program name appended override; e.g., if "`latex`" is the name of the program being run, then `TEXINPUTS.latex` will override `TEXINPUTS`.
2. A program-specific configuration file, for example, a line "`S /a:/b`" in dvips' `config.ps`.
3. A Kpathsea configuration file `texmf.cnf`, containing a line like "`TEXINPUTS=/c:/d`" (see below).
4. The compile-time default.

You can see each of these values for a given search path by using the debugging options (see "Debugging actions" on p. 17).

Config files

Kpathsea reads *runtime configuration files* named `texmf.cnf` for search path and other definitions. The search path used to look for these files is named `TEXMFCNF` (by default such a file lives in the `share/texmf/web2c` subdirectory). *All* `texmf.cnf` files in the search path will be read and definitions in earlier files override those in later files. Thus, with a search path of "`.:$TEXMF`", values from `./texmf.cnf` override those from `$TEXMF/texmf.cnf`.

While reading the description of the format of the file `texmf.cnf` below, please also refer to p. 21, which lists the `texmf.cnf` file on the CD-ROM.

- Comments start with "`%`" and continue to the end of the line.
- Blank lines are ignored.
- A \ at the end of a line acts as a continuation character, i.e., the next line is appended. Whitespace at the beginning of continuation lines is not ignored.
- Each remaining line must look like

      variable[*.progname*] [=] *value*

  where the "=" and surrounding whitespace is optional.
- The *variable* name may contain any character other than whitespace, "=", or ".", but sticking to "`A-Za-z_`" is safest.
- If "`.progname`" is present, the definition only applies if the program that is running is named *progname* or *progname*`.exe`. This allows different flavors of TeX to have different search paths, for example.
- *value* may contain any characters except "`%`" and "`@`". The "`$var.prog`" feature is not available on the right-hand side; instead, you must use an additional variable (see the definition of the variable `latex2e_inputs` for example). A "`;`" in *value* is translated to "`:`" if running under Unix; this is useful to write a single `texmf.cnf` which can be used under both Unix and NT.
- All definitions are read before anything is expanded, so you can use variables before they are defined.

A configuration file fragment illustrating most of these points is shown below:

```
% TeX input files -- i.e.,
% anything found by \input or \openin ...
latex209_inputs = \
 .:$TEXMF/tex/latex209//:$TEXMF/tex//
latex2e_inputs  = \
 .:$TEXMF/tex/latex//:$TEXMF/tex//
```

```
TEXINPUTS = .:$TEXMF/tex//
TEXINPUTS.latex209 = $latex209_inputs
TEXINPUTS.latex2e = $latex2e_inputs
TEXINPUTS.latex = $latex2e_inputs
```

Path expansion

Kpathsea recognizes certain special characters and constructions in search paths, similar to that in Unix shells. As an general example, the following complex path: `~$USER/{foo,bar}//baz` expands to all subdirectories under directories `foo` and `bar` in `$USER`'s home directory that contain a directory or file `baz`. These expansions are explained in the sections below.

Default expansion

If the highest-priority search path (see "Path sources" on p. 12) contains an *extra colon* (i.e., leading, trailing, or doubled), Kpathsea inserts at that point the next-highest-priority search path that is defined. If that inserted path has an extra colon, the same happens with the next-highest. For example, given an environment variable setting

`setenv TEXINPUTS /home/karl:`

and a `TEXINPUTS` value from `texmf.cnf` of

`.:$TEXMF//tex`

then the final value used for searching will be:

`/home/karl:.:$TEXMF//tex`

Since it would be useless to insert the default value in more than one place, Kpathsea changes only one extra "`:`" and leaves any others in place: it checks first for a leading "`:`", then a trailing "`:`", then a doubled "`:`".

Brace expansion

A useful feature is brace expansion, which means that, for instance, `v{a,b}w` expands to `vaw:vbw`. Nesting is allowed. This can be used to implement multiple TeX hierarchies, by assigning a brace list to `$TEXMF`.

For example, in `texmf.cnf`, you find the following definition:

`texdir = $TEXMFLOCAL/tex,!!$TEXMFMAIN/tex`

Then you can write something like:

`TEXINPUTS = .;$texdir//`

which means that after looking in the current directory, first the full `$TEXMFLOCAL/tex` directory tree (on disk) and then the `!!$TEXMFMAIN/tex` tree (using the data base file `ls-R` *only*) will be searched. It is a convenient way for running two parallel TeX structures, one "frozen" (like on a CD-ROM) and the other being continuously updated with new versions as they become available. By using the `$texdir` variable in all definitions, one is sure to always search the up-to-date tree first.

Subdirectory expansion

Two or more consecutive slashes in a path element following a directory *d* is replaced by all subdirectories of *d*: first those subdirectories directly under *d*, then the subsubdirectories under those, and so on. At each level, the order in which the directories are searched is *unspecified*.

If you specify any filename components after the "`//`", only subdirectories with matching components are included. For example, "`/a//b`" expands into directories `/a/1/b`, `/a/2/b`, `/a/1/1/b`, and so on, but not `/a/b/c` or `/a/1`.

Multiple "`//`" constructs in a path are possible, but "`//`" at the beginning of a path is ignored.

List of special characters and their meaning: a summary

The following list summarises the meaning of special characters in Kpathsea configuration files.

`:` Separator in path specification; at the beginning or the end of a path it substitutes the "default" path expansion.
`;` Separator on non-Unix systems (acts like `:`).
`$` Variable expansion.
`~` Represents the user's home directory.
`{...}` Brace expansion, e.g., `a{1,2}b` will become `a1b:a2b`.
`//` Subdirectory expansion. It can occur in the middle or at the end of a path (not at the beginning).
`%` Start of comment.
`\` Continuation character (allows multi-line entries).
`!!` Search *only* database to locate file, *do not* search the disk.

4.2   Filename databases

Kpathsea goes to some lengths to minimize disk accesses for searches. Nevertheless, at installations with enough directories, searching each possible directory for a given file can take an excessively long time (this is especially true if many hundreds of font directories have to be traversed.) Therefore, Kpathsea can use an externally-built "database" file named `ls-R` that maps files to directories, thus avoiding the need to exhaustively search the disk.

A second database file `aliases` allows you to give additional names to the files listed in `ls-R`. This can be helpful to adapt to "8.3" filename conventions in source files.

`ls-R` *filename database*

As explained above, the name of the main filename database must be `ls-R`. You can put one at the root of each TEX installation hierarchy you wish to search (`$TEXMF` by default); most sites have only one hierarchy. Kpathsea looks for `ls-R` files along the `TEXMFDBS` path.

The recommended way to create and maintain "`ls-R`" is to run the `MakeTeXls-R` script coming with the distribution. It is invoked by the various "`MakeTeX...`" scripts. In principle, this script just runs the command

`cd /your/texmf/root && ls -LAR ./ >ls-R`

presuming your system's `ls` produces the right output format (GNU's `ls` is all right). To ensure that the database is always up to date, it is easiest to rebuild it regularly via `cron`, so that for changes in the installed files—perhaps after installing or updating a LATEX package—the file `ls-R` is automatically updated.

If a file is not found in the database, by default Kpathsea goes ahead and searches the disk. If a particular path element begins with "`!!`", however, *only* the database will be searched for that element, never the disk.

`kpsewhich`: *Standalone path searching*

The `kpsewhich` program exercises path searching independent of any particular application. This can be useful as a sort of `find` program to locate files in TEX hierarchies (this is used heavily in the distributed "`MakeTeX...`" scripts).

`kpsewhich` *option... filename...*

Options can start with either "`-`" or "`--`", and any unambiguous abbreviation is accepted.

Kpathsea looks up each non-option argument on the command line as a filename, and returns the first file found. There is no option to return all the files with a particular name (you can run the Unix "`find`" utility for that).

The more important options are described next.

-dpi=*num*   Set the resolution to *num*; this only affects "`gf`" and "`pk`" lookups. "`-D`" is a synonym, for compatibility with `dvips`. Default is 600.

-format=*name*

Set the format for lookup to *name*. By default, the format is guessed from the filename. In fact, the recognized filename extensions and the allowable *name*s (including any leading "`.`") are the same.

You can also specify an integer for *name*; this is the only way to specify formats that don't have an associated suffix, such as MetaPost support files and `dvips` configuration files. It's also somewhat faster, since no unused formats need to be initialized. The integers appear in the output of "`-help`". Currently recognized file type numbers, with their description, possible file extensions, and the corresponding environment variables (between parentheses[1]) as follows:

```
0  Generic font files
   .gf (GFFONTS, GLYPHFONTS,
        TEXFONTS)
1  packed font files
   .pk (PKFONTS, TEXPKS,
        GLYPHFONTS, TEXFONTS)
2  TEX bitmap font
        (GLYPHFONTS)
3  Adobe PostScript font metrics
   .afm (AFMFONTS)
4  METAFONT memory dump
   .base (MFBASES, TEXMFINI)
5  BIBTEX bibliography database
   .bib (BIBINPUTS, TEXBIB)
```

---

[1]You can find definitions for these environment variables in the file `texmf.cnf` (p. 21)

6 BIBTEX styles
   `.bst (BSTINPUTS)`
7 Runtime configuration files
   `.cnf (TEXMFCNF)`
8 Web2c filename database
   `ls-R (TEXMFDBS)`
9 TEX memory dump
   `.fmt (TEXFORMATS, TEXMFINI)`
10 TEX generic font maps
   `.map (TEXFONTMAPS)`
11 MetaPost memory dump
   `.mem (MPMEMS, TEXMFINI)`
12 METAFONT source files
   `.mf  (MFINPUTS)`
13 METAFONT program strings
   `.pool (MFPOOL, TEXMFINI)`
14 METAFONT prettyprinter style files
   `.mft (MFTINPUTS)`
15 MetaPost sources
   `.mp  (MPINPUTS)`
16 MetaPost program strings
   `.pool (MPPOOL, TEXMFINI)`
17 MetaPost support files
   `       (MPSUPPORT)`
18 Ω compiled process
   `.ocp (OCPINPUTS)`
19 Ω font metrics
   `.ofm (OFMFONTS, TEXFONTS)`
20 Ω property list
   `.opl (OPLFONTS, TEXFONTS)`
21 Ω tranlation process files
   `.otp (OTPINPUTS)`
22 Ω virtual fonts
   `.ovf (OVFFONTS, TEXFONTS)`
23 Ω virtual property lists
   `.ovp (OVPFONTS, TEXFONTS)`
24 graphics/figure
   `.eps .epsi (TEXPICTS, TEXINPUTS)`
25 Source input files read by TEX
   `.tex .ltx .dtx .texi .texinfo`
   `.txi .cls .sty .eps .epsi`
   `       (TEXINPUTS)`
26 TEX documentation
   `.ps .pdf .doc .txt (TEXDOCS)`
27 TEX program strings
   `.pool (TEXPOOL, TEXMFINI)`
28 TEX system package sources
   `.dtx .ins (TEXSOURCES)`
29 PostScript header/font
   `.pro (TEXPSHEADERS, PSHEADERS)`
30 Troff fonts
   `       (TRFONTS)`
31 TEX font metric files
   `.tfm (TFMFONTS, TEXFONTS)`
32 PostScript type1 fonts
   `.pfa .pfb (T1FONTS, T1INPUTS,`
   `          TEXPSHEADERS, PSHEADERS)`
33 virtual fonts
   `.vf  (VFFONTS, TEXFONTS)`
34 dvips configuration files
   `config.xxx, xxx.map (TEXCONFIG)`

```
     35 MakeIndex  style files
        .ist   (TEXIDXSTYLE, INDEXSTYLE)
```

These environment variables are set by default in the configuration file `texmf.cnf`. It is only when you want to override one or more of the values specified in that file that you might want to set them explicitly in your execution environment.

Note that the "`-format`" and "`-path`" options are mutually exclusive.

-mode=*string*

Set the mode name to *string*; this also only affects "`gf`" and "`pk`" lookups. No default: any mode will be found.

-must-exist

Do everything possible to find the files, notably including searching the disk. By default, only the `ls-R` database is checked, in the interest of efficiency.

-path=*string*

Search along the path *string* (colon-separated as usual), instead of guessing the search path from the filename. "`//`" and all the usual expansions are supported. The options "`-path`" and "`-format`" are mutually exclusive.

-progname=*name*

Set the program name to *name*. This can affect the search paths via the "`.prognam`" feature in configuration files. The default is "`kpsewhich`".

-show-path=*name*

shows the path used for file lookups of file type *name*. Either a filename extension ("`.pk`", "`.vf`", etc.) or an integer can be used, just as with "`-format`" option.

-debug=*num*

sets the debugging options to *num*.

Examples of use

Let us now have a look at Kpathsea in action.

```
>> kpsewhich -format=.tex article.cls
/usr/local/share/texmf/tex/latex/base/article.cls
```

We are looking for the file `article.cls` in the TEX source file directories (type `.tex`, format type 25). We find it in the subdirectory `tex/latex/base` below the "TEXMF" root directory. To save space, in the following examples we will denote with `...` the repetitive part `/usr/local/share/texmf` preceding each file path.

```
>> kpsewhich tugboat.bib
.../bibtex/bib/beebe/tugboat.bib
```

BIBTEX bibliography databases correspond to format type `.bib`. Here we located file `tugboat.bib`.

```
>> kpsewhich cmr10.pk
.../fonts/pk/ljfour/public/cm/cmr10.600pk
```

```
>> kpsewhich -dpi=300 cmr10.pk
```

```
>> kpsewhich ptmb8r.pk
.../fonts/pk/modeless/dpi597/ptmb8r.pk
```

```
>> kpsewhich -dpi=300 ptmb8r.pk
.../fonts/pk/modeless/dpi300/ptmb8r.pk
```

Font bitmap glyph files of type `.pk` correspond to format type 2. They are used by visualization programs like `dvips` and `xdvi`. On our system we found the Computer Modern file `cmr10` for the mode `ljfour`, at a base resolution of 600 dpi (dots per inch). However, when specifying that we are only interested in a resolution of 300dpi (`-dpi=300`) we are told there is no such font available on the system. In fact, a program like `dvips` or `xdvi` would go off and actually build the `.pk` files at the required resolution using the script `MakeTeXPK`. The last two commands look for a file `ptmb8r.pk`. When specifying no explicit resolution the system returns one (at 597 dpi) which is closest to the "default" set in the `MakeTeXPK` script (600 dpi). However, when specifying the desired resolution (300 dpi) the full path name of the relevant target file is shown.

Next we turn our attention to `dvips`'s header (format type 29) and configuration files (format type 34).

```
>> kpsewhich tex.pro
.../dvips/base/tex.pro
```

```
>> kpsewhich -format=34  psfonts.map
.../dvips/base/psfonts.map
```

```
>> kpsewhich -format=.map config.ps
.../dvips/config/config.ps
```

We first look at a few of the commonly used files, namely the general prolog `tex.pro` for T_EX support, before turning our attention to the generic configuration file (`config.ps`) and the PostScript font map `psfonts.map`. Note how we fool the system by asking for `config.ps` as if it had a suffix of `.map`.

We now look a little closer at the URW Times PostScript support files. The name for these in Berry's font naming scheme is "`utm`". The first file we look at is the configuration file, which contains the name of the map file.

```
>> kpsewhich -format=34  config.utm
.../dvips/config/config.utm
```

The contents of that file is

```
p +utm.map
```

which points to the file `utm.map`, which we want to locate next.

```
>> kpsewhich utm.map
.../dvips/urw/utm.map
```

In this map file, which resides in `dvips`'s `urw` subdirectory, the file names of the Type1 PostScript fonts referenced are defined. The contents looks like (we only show part of the lines):

```
utmb8r NimbusRomNo9L-Medi      ... <utmb8a.pfb
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb
utmr8r NimbusRomNo9L-Regu      ... <utmr8a.pfb
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb
utmbo8r NimbusRomNo9L-Medi "  ... <utmb8a.pfb
utmro8r NimbusRomNo9L-Regu "  ... <utmr8a.pfb
```

Let's, for instance take the Times Regular instance `utmr8a.pfb`, and find its position in the `texmf` directory tree by using a search with format type 32.

```
>> kpsewhich utmr8a.pfb
.../fonts/type1/urw/utm/utmr8a.pfb
```

It should be evident from these few examples how one can easily locate the whereabouts of a given file. This is especially important if you suspect that the wrong version of a file is picked up somehow, since `kpsewhich` will show you the first file encountered.

Debugging actions

Sometimes it is necessary to really investigate how a program resolves file references. To make this feasible in a convenient way Kpathsea offers various debug levels:

  1 `stat` calls (file tests). When running with an up-to-date `ls-R` database this should almost give no output.
  2 References to hash tables (like `ls-R` database, map files, configuration files).
  4 File open and close operations.
  8 General path information for file types searched by Kpathsea. This is useful to find out where a particular path for the file was defined.
 16 Directory list for each path element (only relevant for searches on disk).
 32 File searches.

A value of -1 will set all the above options; in practice you will probably always use these levels if you need any debugging.

Similarly, with the `dvips` program one can, by setting some debug switches, follow in detail where files are picked up from. Alternatively, when a file is not found, the debug trace shows in which directories the program looks for the given file, so that one can get an indication what the problem is.

Generally speaking, as most programs call the Kpathsea library internally, you can select a debug option by using the `KPATHSEA_DEBUG` environment variable, and setting it to (a combination of) values as described in the above list.

Let us consider, as an example, a small LATEX source file, `hello_world.tex`, which contains the following input.

```
\documentclass{article}
\begin{document}
Hello World!
\end{document}
```

This little file only used the font `cmr10`, so let us look how `dvips` prepares the PostScript file.

```
>> dvips -d4100 hello_world -o
```

In this case we have combined `dvips`'s debug class 4 (font paths) with Kpathsea's path element expansion (see `dvips` Reference Manual). We get something like shown below (we have rearranged the output for easier display).

```
debug:start search(file=texmf.cnf, must_exist=1,
                   find_all=1,
  path=.:/usr/local/bin/texlive:/usr/local/bin:
       /usr/local/bin/share/texmf/web2c:/usr/local:
       /usr/local/share/texmf/web2c:
       /.:/.teTeX/TeX/share/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1,
                   find_all=1,
  path=/usr/local/texmf:/usr/local/share/texmf).
kdebug:search(ls-R) =>/usr/local/share/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1,
                   find_all=1,
  path=/usr/local/texmf:/usr/local/share/texmf).
kdebug:search(aliases) =>
kdebug:start search(file=config.ps, must_exist=0,
                   find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//).
kdebug:search(config.ps) =>
  /usr/local/share/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc,
                   must_exist=0, find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//).
kdebug:search($HOME/.dvipsrc) =>
   ...
kdebug:start search(file=psfonts.map, must_exist=0,
                   find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//).
kdebug:search(psfonts.map) =>
  /usr/local/share/texmf/dvips/base/psfonts.map
```

First `dvips` locates its working files. It first found `texmf.cnf` (with the definitions of the paths of the other files), then the file data base `ls-R` (to optimize file searching). It goes on to find the generic configuration file `config.ps`, and then looks for the customization file `.dvipsrc` (which, in this case is *not found*). Finally `dvips` locates the generic map file for PostScript fonts `psfonts.map` (defining the relation between the internal and external names for the PostScript fonts).

At this point `dvips` identifies itself to the user:

```
dvipsk 5.66a Copyright 1986-97 Radical Eye Software
                              (www.radicaleye.com)
```

then goes on to look for the prolog file `texc.pro`,

```
kdebug:start search(file=texc.pro, must_exist=0,
                   find_all=0,
  path=.:/usr/local/texmf/dvips//:
       !!/usr/local/share/texmf/dvips//:
       /usr/local/texmf/fonts//type1//:
       !!/usr/local/share/texmf/fonts//type1//).
kdebug:search(texc.pro) =>
  /usr/local/share/texmf/dvips/base/texc.pro
```

After having found the file, `dvips` outputs date and time, and informs us that it will generate the file `hello_world.ps`, then that it needs the font file `cmr10`, and that the latter is declared as "resident"

```
' TeX output 1997.05.01:1316' -> hello_world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
```

Now the search is on for the file `cmr10.tfm`, which is found, then a few more prolog files (not shown), and finally for the Type1 instance `cmr10.pfb` of the font (which is found) and included in the output file (see last line).

```
kdebug:start search(file=cmr10.tfm, must_exist=1,
                    find_all=0,
  path=.:/usr/local/texmf/fonts/tfm//:
        !!/usr/local/share/texmf/fonts/tfm//:
        /var/tex/fonts/tfm//).
kdebug:search(cmr10.tfm) =>
/usr/local/share/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must_exist=0,
                    find_all=0,
   ...
<texps.pro>.

kdebug:start search(file=cmr10.pfb, must_exist=0,
                    find_all=0,
  path=.:/usr/local/texmf/dvips//:
        !!/usr/local/share/texmf/dvips//:
        /usr/local/texmf/fonts//type1//:
        !!/usr/local/share/texmf/fonts//type1//).
kdebug:search(cmr10.pfb) =>
/usr/local/share/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]
```

4.3   Runtime options

Another of the nice features of Web2c 7.0 is its possibility to control a number of memory parameters (in particular, array sizes) via the runtime file `texmf.cnf` read by Kpathsea. A detailed list of all set-table parameters can be found in that file (see p. 21, Part 3 starting at line 261). The most interesting values are:

main_memory Total words of memory available, for TEX, METAFONT, and MetaPost. You must make a new format file for each different setting. For instance, you could generate a "huge" version of TEX, and call the format file `hugetex.fmt`. Using the standard way of specifying the program name used by Kpathsea the particular value of the `main_memory` variable will then be read from `texmf.cnf` (See p. 21, line 280 for the generic value and line 281 for the "huge" one instantiated by `hugetex`).

extra_mem_bot Extra space for "large" TEX data structures: boxes, glue, breakpoints, etc. Especially useful if you use P₁CTEX.

font_mem_size Number of words for font information available for TEX. This is more or less the total size of all TFM files read.

hash_extra Additional space for the hash table of control sequence names. Approximately 10,000 control sequences can be stored in the main hash table; if you have a large book with numerous cross-references, this might not be enough. On line 297 and 298 of file `texmf.cnf` as shown in p. 21 you see that both the `hugetex` and `pdftex` program invocations ask for an extra 10,000 control sequences (the default value of `hash_extra` is zero, as seen on line 296).

Of course, this facility is no substitute for truly dynamic arrays and memory allocation, but since this is extremely difficult to implement in present TEX, these runtime parameters provide a practical compromise allowing some flexibility.

5   Other packages on the CD-ROM

While the main portion of **TEX Live** (the fonts, macros and documentation) can be used on any TEX system, the set of runnable binaries is not suitable for everyone. To make the disk as widely useful as possible, we have included the original distributions of four complete TEX systems, two for Macintosh, one for Windows 95, and one for DOS and OS/2. Windows 3.1 users should look at the GUTenberg distribution on the CD-ROM.

5.1   OzTEX[2]

OzTEX is a Macintosh TEX system created by Andrew Trevorrow. The OzTEX application includes TEX, INITEX, a DVI previewer, a DVI-to-PostScript translator (Tom Rokicki's `dvips`) and a driver for QuickDraw printers. OzTEX also includes `dvidvi`, `dvicopy`, and Angus Duggan's PostScript utilities: `psbook`, `psnup`, `psselect` and `pstops`.

The version of `dvips` included in OzTEX supports HyperTEX and the partial downloading of PostScript fonts.

---

[2]This section was written by Andrew Trevorrow.

It has also been enhanced for Mac users in a number of ways: Standard Mac PostScript fonts (LWFN files) can be downloaded, fully or partially. All OzTeX-specific `\special` commands are supported, such as the inclusion of PICT/PNTG/EPSF files. The `dvips` output can be sent directly to the current printer.

OzTeX's previewer has lots of features to make it easy to proofread DVI files. It can handle PK and PostScript fonts. Anti-aliasing is supported. Virtual fonts are processed on the fly. The previewer supports most of the `\special` commands generated by LaTeX's `color`, `graphics/x` and `hyperref` packages. It recognizes all `dvips`-specific `\specials` and those it cannot handle (like rotation) are silently ignored.

OzTeX includes all the most popular formats and macro packages. Plain TeX, LaTeX, AMS-TeX, AMS-LaTeX and REVTeX are all installed and ready to run.

OzTeX is easy to extend and customize. A default configuration file is read when OzTeX starts up; it contains a host of parameters for setting up search paths, telling TeX how much memory to allocate for various arrays, specifying which TFMs are for PostScript fonts, etc. A Config menu makes it easy to load other config files at any time. And for even more flexibility, OzTeX can automatically load a specified config file just before typesetting, previewing or printing.

Additional programs

The usual assortment of TeX-related programs are provided with OzTeX, including OzMF, a Mac implementation of METAFONT, and OzMP, a Mac port of John Hobby's MetaPost program for producing PostScript pictures using a METAFONT-like language.

The following programs are also distributed with OzTeX, courtesy of their authors; BibTeX by Vince Darley; MakeIndex by Rick Zaccone; Excalibur, a TeX/LaTeX spelling checker, by Rick Zaccone and Robert Gottshall; and AlphaLite, a TeX/LaTeX-savvy text editor, by Pete Keleher.

For the latest information about OzTeX, keep an eye on the Web page at the URL http://www.kagi.com/authors/akt/oztex.html.

An even better way to keep up-to-date is to join the `oztex-info` mailing list. To subscribe, send some e-mail to

    majordomo@maths.adelaide.edu.au

with the following line in the *body* of the message:

    subscribe oztex-info

OzTeX is distributed as shareware, so you are welcome to try it out before paying the registration fee. The individual fee is US$30 and the site fee is US$300. See the "Shareware Fee" item in OzTeX's Help menu for details on how to pay. E-mail support is provided to registered users. Send all queries and comments to Andrew Trevorrow (akt@kagi.com).

5.2 CMacTeX[3]

CMacTeX is an implementation of TeX for the Macintosh by Thomas Kiffe (mailto:tkiffe@math.tamu.edu). It includes the three main parts of any TeX installation—TeX, METAFONT and dvips. It also includes two dvi previewers, a utility for printing dvi files on a non PostScript printer, a PostScript previewer and numerous utilities for manipulating TeX fonts. Full support for the automatic generation of pk font files is an integral part of the distribution. CMacTeX can be configured to work in an integrated fashion with BBEdit, Alpha, and MPW. It will run on any Macintosh with 8 MB of RAM and System 7.

CMacTeX is shareware. The registration fee is US$35 for a single-user license and US$150 for a site license.

Installation instructions can be found in the file `/macintosh/cmactex/ReadMeFirst`

5.3 MiKTeX[4]

MiKTeX 1.07 is an implementation by Christian Schenk (mailto:cschenk@berlin.snafu.de) of TeX and METAFONT related utilities for Windows NT and Windows 95. The MiKTeX distribution includes TeX; LaTeX 2_ε Dec'96 including standard packages; METAFONT; MetaPost; dvips MakeIndex; BIBTeX; YAP (Yet Another Previewer); TeXware (dvitype etc.); METAFONTware (gftopk etc.); psutils (psselect, pstops etc.); and DVIcopy.

Installation instructions can be found in the file `/win32/miktex/README.TXT`

5.4 emTeX

The emTeX distribution for DOS and OS/2 is wriiten by Eberhard Mattes (mailto:mattes@azu.informatik.uni-stuttgart.de). It includes the TeX typesetter, the METAFONT font generation program, printer drivers, screen pre-

---

[3]This section is taken from the CMacTeX documentation.

[4]This section is drawn from the documentation.

viewers, and tools like BIBTEX and MakeIndex. It also includes the macro packages LATEX 2.09 and LATEX 2ε. Fonts are included as pixel files and METAFONT source files.

   Installation instructions can be found in the file /msdos/emtex/README.ENG

6   The `texmf.cnf` file

```
1    % original texmf.cnf -- runtime path configuration file for kpathsea.
2    % (If you change or delete 'original' on the previous line, the
3    % distribution won't install its version over yours.)
4    % Public domain.
5    %
6    % What follows is a super-summary of what this .cnf file can
7    % contain. Please read the Kpathsea manual for more information.
8    %
9    % texmf.cnf is generated from texmf.cnf.in, by replacing @var@ with the
10   % value of the Make variable 'var', via a sed file texmf.sed, generated
11   % (once) by kpathsea/Makefile (itself generated from kpathsea/Makefile.in
12   % by configure).
13   %
14   % Any identifier (sticking to A-Za-z_ for names is safest) can be assigned.
15   % The '=' (and surrounding spaces) is optional.
16   % No % or @ in texmf.cnf.in, for the sake of autogeneration.
17   % (However, %'s and @'s can be edited into texmf.cnf or put in envvar values.)
18   % $foo (or ${foo}) in a value expands to the envvar or cnf value of foo.
19   %
20   % Earlier entries (in the same or another file) override later ones, and
21   % an environment variable foo overrides any texmf.cnf definition of foo.
22   %
23   % All definitions are read before anything is expanded, so you can use
24   % variables before they are defined.
25   %
26   % If a variable assignment is qualified with '.PROGRAM', it is ignored
27   % unless the current executable (last filename component of argv[0]) is
28   % named PROGRAM.  This foo.PROGRAM construct is not recognized on the
29   % right-hand side. For environment variables, use FOO_PROGRAM.
30   %
31   % Which file formats use which paths for searches is described in the
32   % various programs' and the kpathsea documentation.
33   %
34   % // means to search subdirectories (recursively).
35   % A leading !! means to look only in the ls-R db, never on the disk.
36   % A leading/trailing/doubled : in the paths will be expanded into the
37   %   compile-time default. Probably not what you want.
38
39   %  Part 1: Search paths and directories.
40
41   % The root of everything below.
42   prefix = $SELFAUTOPARENT
43
44   % You can set an environment variable to override this if you're testing
45   % a new TeX tree, without changing anything else.
46   %
47   % You may wish to use one of the $SELFAUTO... variables here so TeX will
48   % find where to look dynamically.  See the manual and the definition
49   % below of TEXMFCNF.
50   %
51   % If you have multiple trees, you can use shell brace notation, for example:
52   % TEXMF = {/usr/local/mytex,/usr/local/othertex}
53   % and also set TEXMFDBS to /usr/local/mytex:/usr/local/othertex.
54   TEXMFLOCAL=/usr/local/texmf
55   TEXMFMAIN = $prefix/share/texmf
```

```
56   TEXMFLS_R = $TEXMFLOCAL
57   % Where to look for ls-R files.  There need not be an ls-R in the
58   % directories in this path, but if there is one, Kpathsea will use it.
59   TEXMFDBS = $TEXMFLOCAL;$TEXMFMAIN
60
61   % Where you want generated files to go. Choose one of the texmf trees
62   % listed in $TEXMF. The following is the default:
63   VARTEXMF = $TEXMFLOCAL
64
65   % The TeX inputs and fonts directories.
66   texdir = {$TEXMFLOCAL/tex,!!$TEXMFMAIN/tex}
67   omegadir ={$TEXMFLOCAL/omega,!!$TEXMFMAIN/omega}
68   fontdir = {$TEXMFLOCAL/fonts,!!$TEXMFMAIN/fonts}
69   dbmain=!!$TEXMFMAIN
70   %
71   dbtex = $texdir
72   dbomega = $omegadir
73   dbfonts = $fontdir
74
75   % TeX input files -- i.e., anything to be found by \input or \openin,
76   % including .sty, .eps, etc.
77   TEXINPUTS = .;$dbtex//
78
79   % LaTeX 2e specific macros are stored in latex.
80   % latex209 is not supported, at the request of the authors of LaTeX
81   %
82   latex_inputs = .;$dbtex/latex//;$dbtex/generic//
83   TEXINPUTS.latex = $latex_inputs
84   TEXINPUTS.latextex = $latex_inputs
85
86   % Omega
87   TEXINPUTS.lambda2e = .;$dbomega/lambda//;$latex_inputs
88   TEXINPUTS.lambda = .;$dbomega/lambda//;$latex_inputs
89
90   % Fontinst needs to read afm files.
91   TEXINPUTS.fontinst = .;$dbtex//;$dbfonts/afm//
92
93   % Plain TeX.  Have the command tex check all directories as a last
94   % resort, we may have plain-compatible stuff anywhere.
95   plain_inputs = $dbtex/plain//;$dbtex/generic//
96   TEXINPUTS.textex = .;$dbtex//;$plain_inputs
97   TEXINPUTS.tex = .;$plain_inputs;$dbtex//
98   TEXINPUTS.omega = .;$dbomega//;$plain_inputs;$dbtex//
99
100  % INITEX. May as well make this separate so it can search on disk;
101  % initex is seldom run, and might be used directly after files have been
102  % added, when ls-R may not up be to date.
103  TEXINPUTS.initex = .;$texdir//
104  TEXINPUTS.iniomega = .;$omegadir//;$texdir//
105
106  % Earlier entries override later ones, so put this last.
107  TEXINPUTS.viromega = .;$dbomega//;$dbtex//
108  TEXINPUTS = .;$dbtex//
109
110  % Metafont, MetaPost inputs.
111  MFINPUTS = .;{$TEXMFLOCAL,$dbmain}/metafont//;{$dbfonts,$VARTEXFONTS}/source//
112  MPINPUTS = .;{$TEXMFLOCAL,$dbmain}/metapost//
113
114  % Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see
115  % web2c/INSTALL), and string pools (.pool) for ini{tex,mf,mp}.  It is
116  % silly that we have six paths and directories here (they all resolve to
```

```
117  % a single place by default), but historically ...
118  TEXFORMATS = .;{$TEXMFLOCAL,$dbmain}/web2c
119  MFBASES = .;{$TEXMFLOCAL,$dbmain}/web2c
120  MPMEMS = .;{$TEXMFLOCAL,$dbmain}/web2c
121  TEXPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
122  MFPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
123  MPPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
124
125  % If you have a read-only central font directory and therefore need to
126  % cache MakeTeXPK-created fonts locally, set this as appropriate and
127  % also set the `vartexfonts' feature in MT_FEATURES in MakeTeXnames.cnf.
128  VARTEXFONTS = /var/tex/fonts
129
130  % Device-independent font metric files.
131  VFFONTS = .;$dbfonts/vf//
132  TFMFONTS = .;$dbfonts/tfm//;$VARTEXFONTS/tfm//
133
134  % The $MAKETEX_MODE below means the drivers will not use a cx font when
135  % the mode is ricoh.  If no mode is explicitly specified, kpse_prog_init
136  % sets MAKETEX_MODE to /, so all subdirectories are searched.  See the manual.
137  PKFONTS = .;{$dbfonts,$VARTEXFONTS}/pk/$MAKETEX_MODE//
138
139  % xdvi needs to find bitmaps for PostScript fonts, which can be
140  % generated by the standalone program gsftopk, among others.
141  PKFONTS.xdvi = .;{$dbfonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modeless}//
142
143  % Similarly for the GF format, which only remains in existence because
144  % Metafont outputs it (and MF isn't going to change).
145  GFFONTS = .;$dbfonts/gf/$MAKETEX_MODE//
146
147  % A backup for PKFONTS and GFFONTS. Not used for anything.
148  GLYPHFONTS = .;$dbfonts
149
150  % For texfonts.map and included map files used by MakeTeXPK.
151  % See ftp;//ftp.tug.org/tex/fontname.tar.gz.
152  TEXFONTMAPS = .;{$TEXMFLOCAL,$dbmain}/fontname
153
154  % BibTeX bibliographies and style files.
155  BIBINPUTS = .;{$TEXMFLOCAL,$dbmain}/bibtex/bib//
156  BSTINPUTS = .;{$TEXMFLOCAL,$dbmain}/bibtex/bst//
157
158  % MFT style files.
159  MFTINPUTS = .;{$TEXMFLOCAL,$dbmain}/mft//
160
161  % PostScript headers, prologues (.pro), encodings (.enc) and fonts.
162  TEXPSHEADERS = .;$TEXMFLOCAL/dvips;$dbmain/dvips//;$dbfonts/type1//
163
164  % PostScript Type 1 outline fonts.
165  T1FONTS = .;$dbfonts/type1//;$TEXMFLOCAL/dvips//;$dbmain/dvips//
166
167  % PostScript AFM metric files.
168  AFMFONTS = .;$dbfonts/afm//
169
170  % Dvips' config.* files (this name should not start with `TEX'!).
171  TEXCONFIG = .;$TEXMFLOCAL/dvips;$dbmain/dvips//
172
173  % Makeindex style (.ist) files.
174  INDEXSTYLE = .;{$TEXMFLOCAL,$dbmain}/makeindex//
175
176  % Used by DMP (ditroff-to-mpx), called by makempx -troff.
177  TRFONTS = /usr/lib/font/devpost
```

```
178  MPSUPPORT = .;{$TEXMFLOCAL,$dbmain}/metapost/support
179
180  % For xdvi to find mime.types and .mailcap, if they do not exist in
181  % $HOME.  These are single directories, not paths.
182  % (But the default mime.types, at least, may well suffice.)
183  MIMELIBDIR = $prefix/etc
184  MAILCAPLIBDIR = $prefix/etc
185
186  % TeX documentation and source files, for use with kpsewhich.
187  TEXDOCS = .;{$TEXMFLOCAL,$dbmain}/doc//
188  TEXSOURCES = .;{$TEXMFLOCAL,$dbmain}/source//
189
190  % Omega-related fonts and other files.
191  OFMFONTS = .;{$dbfonts,$VARTEXFONTS}//ofm//
192  OPLFONTS = .;{$dbfonts,$VARTEXFONTS}//opl//
193  OVFFONTS = .;{$dbfonts,$VARTEXFONTS}//ovf//
194  OVPFONTS = .;{$dbfonts,$VARTEXFONTS}//ovp//
195  OTPINPUTS = .;{$TEXMFLOCAL,$dbmain}/omega/otp//
196  OCPINPUTS = .;{$TEXMFLOCAL,$dbmain}/omega/ocp//
197
198  % For MakeTeX.common, MakeTeX.site, ls-R.
199  web2cdir = {$TEXMFLOCAL,$dbmain}/web2c
200  TEXMFCNF_DIR = $TEXMFMAIN/web2c
201
202  % The MakeTeX* scripts rely on KPSE_DOT. Do not change it.
203  KPSE_DOT = .
204
205  % This definition isn't used from this .cnf file itself (that would be
206  % paradoxical), but the compile-time default in paths.h is built from it.
207  % The SELFAUTO* variables are set automatically from the location of
208  % argv[0], in kpse_set_progname.
209  %
210  % About the /. construction;
211  % 1) if the variable is undefined, we'd otherwise have an empty path
212  %    element in the compile-time path.  This is not meaningful.
213  % 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
214  %    which would search the entire world.
215  %
216  % The TETEXDIR stuff isn't likely to relevant unless you're using teTeX,
217  % but it doesn't hurt.
218  %
219  TEXMFCNF =.;$SELFAUTOLOC;$SELFAUTODIR;$SELFAUTODIR/share/texmf/web2c;$SELFAUTOPARENT;\
220  $SELFAUTOPARENT/share/texmf/web2c;/.$TETEXDIR;/.{$TEXMFLOCAL,$dbmain}/web2c;$web2cdir
221
222  %  Part 2; Non-path options.
223
224  % Write .log/.dvi/etc. files here, if the current directory is unwritable.
225  % TEXMFOUTPUT = /tmp
226
227  % If a dynamic file creation fails, log the command to this file, in
228  % either the current directory or TEXMFOUTPUT.  Set to the
229  % empty string or  0  to avoid logging.
230  MISSFONT_LOG = missfont.log
231
232  % Set to a colon-separated list of words specifying warnings to suppress.
233  % To suppress everything, use TEX_HUSH = all; this is equivalent to
234  % TEX_HUSH = checksum;lostchar;readable;special
235  TEX_HUSH = 0
236
237  % Enable system commands via \write18{...}?
238  shell_escape = 0
```

```
239
240  % Allow TeX \openout on filenames starting with '.' (e.g., .rhosts)?
241  openout_any = 0
242
243  % Enable the MakeTeX... scripts by default?  These must be set to 0 or 1.
244  % Particular programs can and do override these settings, for example
245  % dvips's -M option.  Your first chance to specify whether the scripts
246  % are invoked by default is at configure time.
247  %
248  % These values are ignored if the script names are changed; e.g., if you
249  % set DVIPSMAKEPK to 'foo', what counts is the value of the environment
250  % variable/config value 'FOO', not the 'MAKETEXPK' value.
251  %
252  % MAKETEXTEX = 0
253  % MAKETEXPK = 0
254  % MAKETEXMF = 0
255  % MAKETEXTFM = 0
256
257  % What MetaPost runs to make MPX files.  This is passed an option -troff
258  % if MP is in troff mode.  Set to '0' to disable this feature.
259  MPXCOMMAND = makempx
260
261  %  Part 3; Array and other sizes for TeX (and Metafont and MetaPost).
262  %
263  % If you want to change some of these sizes only for a certain TeX
264  % variant, the usual dot notation works, e.g.,
265  % main_memory.hugetex = 20000000
266  %
267  % If a change here appears to be ignored, try redumping the format file.
268
269  % Memory. Must be less than 8,000,000.
270  %
271  % main_memory is relevant only to initex, extra_mem_* only to non-ini.
272  % Thus, have to redump the .fmt file after changing main_memory; to add
273  % to existing fmt files, increase the other.  (To get an idea of how
274  % much, try \tracingstats=2 in your TeX source file;
275  % web2c/tests/memtest.tex might also be interesting.)
276  %
277  % To increase space for boxes (as might be needed by, e.g., PiCTeX),
278  % increase extra_mem_bot.
279  %
280  main_memory = 263000 % words of memory available; also applies to mf&mp
281  main_memory.hugetex = 1100000
282  extra_mem_top = 0     % extra high memory for chars, tokens, etc.
283  extra_mem_bot = 0     % extra low memory for boxes, glue, breakpoints, etc.
284
285  % Words of font info for TeX (total size of all TFM files, approximately).
286  font_mem_size = 100000
287  font_mem_size.hugetex = 400000
288  font_mem_size.pdftex = 400000
289
290  % Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
291  font_max = 500
292  font_max.hugetex = 900
293
294  % Extra space for the hash table of control sequences (which allows 10K
295  % names as distributed).
296  hash_extra = 0
297  hash_extra.hugetex = 10000
298  hash_extra.pdftex = 10000
299
```

```
300  % Max number of characters in all strings, including all error messages,
301  % help texts, font names, control sequences.  These values apply to TeX and MP.
302  pool_size = 125000
303  pool_size.hugetex = 500000
304  pool_size.pdftex = 500000
305  % Minimum pool space after TeX/MP's own strings; must be at least
306  % 25000 less than pool_size, but doesn't need to be nearly that large.
307  string_vacancies = 25000
308  string_vacancies.hugetex = 45000
309  string_vacancies.pdftex = 45000
310  max_strings = 15000            % max number of strings
311  max_strings.hugetex = 55000              % max number of strings
312  max_strings.pdftex = 55000               % max number of strings
313  pool_free = 5000               % min pool space left after loading .fmt
314
315  % Hyphenation trie. As distributed, the maximum is 65535; this should
316  % work unless 'unsigned short' is not supported or is smaller than 16
317  % bits.  This value should suffice for UK English, US English, French,
318  % and German (for example).  To increase, you must change
319  % 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
320  % the trie will then consume four bytes per entry, instead of two.
321  %
322  % US English, German, and Portuguese; 30000.
323  % German; 14000.
324  % US English; 10000.
325  %
326  trie_size = 64000
327
328  hyph_size = 1000         % number of hyphenation exceptions, >610 and <32767.
329  buf_size = 3000          % max length of input lines or control sequence names
330  nest_size = 100          % simultaneous semantic levels (e.g., groups)
331  max_in_open = 15         % simultaneous input files and error insertions
332  param_size = 500         % simultaneous macro parameters
333  save_size = 4000         % for saving values outside current group
334  save_size.pdftex = 30000        % for saving values outside current group
335  save_size.hugetex = 30000       % for saving values outside current group
336  stack_size = 300         % simultaneous input sources
337
338  % These work best if they are the same as the I/O buffer size, but it
339  % doesn't matter much.  Must be a multiple of 8.
340  dvi_buf_size = 16384 % TeX
341  gf_buf_size = 16384  % MF
342
343  % It's probably inadvisable to change these. At any rate, we must have:
344  % 45 < error_line       < 255;
345  % 30 < half_error_line < error_line - 15;
346  %      max_print_line  < 60 ;
347  % These apply to Metafont and MetaPost as well.
348  error_line = 79
349  half_error_line = 50
350  max_print_line = 79
```

7 Catalogue of Packages

Table 1: **TEX Live** packages

Package	*Collection*	*Description*

IV   The UK TeX Users' Group

The 1995–96 UKTUG committee

R. Fairbairns	Chair
P. Abbott	Treasurer and Membership Secretary
D. P. Carlisle	Committee Secretary
M. Clark	Meetings Secretary

K. Bazargan; S. P. Q. Rahtz; M. D. Wooding.

Book Discounts for UKTUG members

We have arrangements with Addison-Wesley for their well-known TeX-related publications, and with International Thomson Publishing to supply any of the very excellent O'Reilly & Associates Inc. series of books to members.

The agreed list of books, together with the discounted (at least 20%) price, is distributed occasionally with *Baskerville*, but is always available from the Treasurer, Peter Abbott. The quoted price includes the cost of postage and packing.

We are only allowed to offer this service to **current** members of the UK TeX Users' Group and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott (address in *Baskerville* masthead). Make cheques payable to 'UKTUG' please. Books from Addison-Wesley are delivered direct but books from O'Reilly will be routed through UKTUG. *In all cases* please notify Peter Abbott by email, phone, fax or letter when books are delivered. This service is unfortunately not a speedy process

V Obtaining TEX

edited by Peter Abbott

From the network – CTAN

The UK TEX Archive on `ftp.tex.ac.uk` is part of the CTAN (Comprehensive TEX Archive Network) collaborating network of archives on the Internet organised by the TEX Users Group.

The CTAN archives run an enhanced *ftp* server which supports dynamic compression, uncompression, and archive creation options. Fetch the top-level file `README.archive-features` for information. The server also supports site-defined commands to assist you. Please read `README.site-commands` for a brief overview.

Please report any problems with CTAN archives via email to `ctan@urz.Uni-Heidelberg.de`.

The main directories which make up CTAN are listed below; readers are referred to Graham Williams' *TEX and LATEX Catalogue* which is available from CTAN as `help/Catalogue/catalogue.html`

biblio bibliography-related files, such as BIBTEX.

digests back issues of TEX-related periodicals

dviware contains the various `dvi`-to-whatever filters and drivers

fonts fonts, both sources and pre-compiled

graphics utilities and macros related to graphics

help overviews of the archive and the TEX system

info files and tutorials which document various aspects of TEX

indexing utilities and related files for indexing

language material for typesetting non-English documents

macros macros packages for TEX and style files

support programs which can be used in support of TEX

systems complete system setups, organized by operating system

tools the various archiving tools used on CTAN

web contains WEB-related files and utilities

Unix – CD-ROM

GUTenberg and UKTUG, in collaboration with TUG and NTG, have produced a plug-and-play CD-ROM based on Thomas Esser's teTEX distribution. As it uses the ISO 9660 standard, the platform-independent files can, in principle, be read on all operating systems which are compatible with that format.

Unix executables for the following platform/operating system combinations are included: Digital alpha-osf (2.0 and 3.2), Hewlett Packard hpux (9.01 and 10.01), Intel i386 bsdi2.0, freebsd (2.0.5 and 2.1.0) netbsd (1.0 and 1.1), Intel i486 (linux and linuxaout), m68k (linux, linuxoldld, and nextstep3), mips (irix 5.2, 5.3 and ultrix4.4) IBM RS6000 (aix3.2 and aix4.1.1) Sparc Solaris (2.4 and 2.5) and Sunos 4.1.3.

For full details see the article in *Baskerville* 6.2.

The CD is available to members of TEX user groups at £15 and to non-members at £25. Order the disk from Peter Abbott; see the section 'PC and Mac disks' for details.

DOS – CD-ROM

UKTUG distributes the comprehensive 4AllTEX CD-ROM, created by the Dutch TEX Users' Group (NTG), now in its 3rd edition. This costs £25 for 2 CDs, and is for DOS users.

PC and Mac disks

The UKTUG distributes an emTEX kit for PCs, and an OzTEX kit for Macintosh. The cost covers copying and postage costs, together with the shareware fee for OzTEX (and other Mac programs) and Eddi4TEX. Each set costs £30, and is available from Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Cheques must be payable to 'UKTUG'. Please note that this service *is available to UKTUG members only*. Each set comes with an installation guide, and (at least) full TEX and METAFONT, a previewer, a PostScript driver, and CM fonts. Two update disks a year will be sent out automatically, with the current version of LATEX 2ε, and other goodies. A subscription service is