A Directory Structure for TEX Files
Version 0.98

TUG Working Group on a TEX Directory Structure (TWG-TDS)

June 14, 1995

This is a draft specification.
This is only a draft.

This specification may be referred to as the "TWG-TDS 0.98" or "TWG-TDS" for short, or just "TDS" when the context makes it clear which TEX directory structure is meant.

Please direct all questions, suggestions, or bug reports, to `twg-tds@shsu.edu` or by mail to Norman Walsh, O'Reilly & Associates, Inc., 90 Sherman Street, Cambridge, MA 02140, USA.

This document is available in LATEX, `DVI`, POSTSCRIPT, Texinfo, HTML, and other formats from `ftp://jasper.ora.com/pub/twg-tds/*` and from `/tex-archive/tds` on any CTAN host.

Contents

1 Introduction **3**
- 1.1 The Role of the TDS 3
- 1.2 Conventions 3

2 Basics **4**
- 2.1 Subdirectory Searching 4
- 2.2 Constraints 5
- 2.3 Rooting the Tree 6
- 2.4 Top Level Directories 6

3 Macros **7**

4 Fonts **8**
- 4.1 Font Bitmaps 9

5 Non-font METAFONT Files **10**

6 METAPOST **11**

7 BibTeX **11**

8 Documentation **11**

9 Package Distribution **13**

10 Summary **15**

A Implementation Issues **15**
- A.1 Adoption of the TDS 16
- A.2 Expediting the Process 16
- A.3 Recursive Subdirectory Searching 17
- A.4 Subdirectory Searching Syntax 17
- A.5 Mixed Case Filenames 17

B Is There a Better Way? **18**
- B.1 A Better Macro Structure 18
- B.2 A Better Font Structure 18

C Related References **19**

D About the Committee **20**

1 Introduction

TeX is a powerful, flexible typesetting system used by thousands of people around the world. It is extremely portable and runs on dozens of operating systems. One unfortunate consequence of TeX's flexibility is that there is no single "right" way to install it. This has resulted in hundreds of different installed configurations.

The primary purpose of this document is to describe a standard TeX Directory Structure (TDS) for macros, fonts, and other such implementation-independent TeX files. As a matter of practicality, this document also suggests ways to incorporate the rest of the TeX files into a single structure. In the not-so-long run a consistent directory structure will make it much easier to install and maintain TeX. We hope that administrators and developers of both free and commercial implementations of TeX will adopt this standard. It has been designed to work on all modern systems. In particular, this Technical Working Group (TWG) believes it is usable under Unix, MS-DOS, OS/2, MacOS, and VMS.

This document is intended both for the TeX system administrator at a site and for people preparing TeX distributions—everything from a complete runnable system to a single macro or style file. It will also help TeX users find their way around systems organized this way.

1.1 The Role of the TDS

The role of the TDS is to stabilize the organization of TeX-related software packages that are installed and in use, possibly on multiple platforms simultaneously.

At first glance, it may seem that the CTAN archives fulfill (at least) part of this role, but this is not the case. The role of the CTAN archives is to simplify *archiving* and *retrieval* of packages, not their installation and use.

In fact, the roles of the TDS and CTAN are frequently in conflict, as you will discover as you read this document. For distribution in CTAN, it's useful to combine many different types of files into a single unit; for use, it is traditional to segregate files (even similar files) from a single package into separate, occasionally distant, directories.

1.2 Conventions

In this document, "/" is used to separate filename components; for example, `texmf/fonts`. This is the Unix convention but the ideas are in no way Unix-specific.

In this document, "TeX" generally means "the TeX system, including METAFONT, DVI drivers, utilities, etc.," not just the TeX program itself.

Packages, as described in this document, are *not* LaTeX 2ε packages, even though there sometimes exist LaTeX packages with the same names. LaTeX packages are style files that add more macro code to document classes. TDS packages are sets of files distributed, installed, and maintained as units.

The following typographic conventions are used:

`filename` Filenames are typeset in typewriter (constant width) type.

`command` Commands are typeset in italics.

`literal` Literal text is typeset in typewriter type.

`replaceable` Replaceable text is typeset in typewriter oblique. Replaceable text, such as *`package`*, identifies a class of things.

2 Basics

2.1 Subdirectory Searching

Many TEX installations currently store large numbers of related files in single directories. Frequently, for example, all `TFM` files go in a single directory and most, if not all, TEX input files (LATEX style files, Plain TEX macro files, etc.) go in another single directory.

This monolithic arrangement makes maintenance of a TEX system difficult: it is difficult to determine what files are used by which packages, what files need to be updated when a new version is installed, or what files should be deleted if a package is removed. It can also be the source of catastrophic error if two or more packages happen to have input files with the same name.

Therefore, the TWG felt each package should be in a separate directory. But we recognized that explicitly listing all directories to be searched would be unbearable. There are dozens of packages a site may wish to install. Aside from anything else, listing that many directories would lead to paths many thousands of characters long, overflowing the available space on some systems.

Also, if all directories are explicitly listed, installing or removing a new package would mean changing a path as well as installing or removing the actual files. This would be a time-consuming and error-prone operation, even with implementations that provide some ability to specify the directories to search at runtime (in a `TEXINPUTS` environment variable or a system configuration file, for example). On systems without any form of runtime configuration, it would require recompiling software, an intolerable burden.

As a result, the TWG concluded that a comprehensive TDS required that implementations of TEX must support some form of implicit subdirectory searching. In other words, it must be possible to specify that TEX, METAFONT, and their companion utilities search in both a specified directory and recursively through all subdirectories of that directory when looking for an input file. We encourage implementors to provide subdirectory searching (at the option of the installer and user) for all paths.

The TDS does not specify the order in which subdirectories are descended. It follows that two identically named files in a search path leads to ambiguity. The TDS does not define which one is found. In order to resolve this ambiguity, a search path must be specified that lists the respective subtrees in the desired search order.

The TDS does not specify a syntax for specifying recursive searching, but we encourage implementors to provide interoperability wherever possible. See Section A.4, “Subdirectory Searching Syntax” for a list.

2.2 Constraints

Having accepted the necessity for subdirectory searching, the TWG turned its attention next to other constraints placed on the TDS. The other constraints arise because it must be possible to use the TDS on any platform.

2.2.1 Alphabetic Case in Filenames

Many, but not all, operating systems disregard the distinction between upper- and lowercase letters in filenames. The ISO-9660 format (described below) does not allow mixed-case names. The TDS does *not* require monocase names, although it is clear that mixed-case filenames may cause problems in some environments. Filenames which are identical except with respect to case are particularly likely to be troublesome.

2.2.2 CD-ROMs

One use of the TDS will be the creation of mountable generic TEX-related distributions on CD-ROM. CD-ROM vendors will want to use ISO-9660 format because it is the only universally acceptable file system format for CD-ROMs. ISO-9660 is portable, but it imposes strict limitations on file and directory names:

- File and directory names, not including any directory path or extension part, may not exceed eight characters.
- Filenames may have a single extension. Extensions may not exceed three characters.
- Names and extensions may consist of *only* the characters `A`–`Z`, `0`–`9`, and underscore.
- A period separates the filename from the extension. It is always present, even if the name or extension is missing (e.g., `FILENAME.`, `.EXT`).
- A version number, ranging from 1–32767, is found at the end of the file extension, separated by a semicolon (e.g., `FILENAME.EXT;1`).
- Only eight directory levels are allowed, including the top-level (mounted) directory. Thus, the deepest valid TDS path is:

  ```
  texmf/L2/L3/L4/L5/L6/L7/L8/FOO.BAR;1
  1     2  3  4  5  6  7  8
  ```

  Here is the deepest TDS example, which needs only seven levels:

  ```
  texmf/fonts/pk/cx/public/cm/dpi300/cmr10.pk
  1     2     3  4         5  6  7
  ```

Note: Some systems display a modified format of ISO-9660 names, mapping alphabetic characters to lowercase, removing version numbers and trailing periods, etc. This does not affect the format of names on the CD-ROM itself.

2.3 Rooting the Tree

The root of the TDS should be a directory containing only TeX-related materials. In this document, we shall designate this directory '`texmf`', but the actual name of the directory is up to the installer. On PC networks, for example, this could map to a logical drive specification such as `T:`. We recommend naming the directory of the system-wide TeX installation `texmf`, where possible.

Similarly, the location of this directory within the file system is site-dependent. On many systems, this may be at the root of the file system; on Unix systems, it would traditionally be located in `/usr/local` or `/usr/local/lib`.

The name `texmf` was chosen for several reasons: it reflects the fact that the directory contains files to an entire TeX system (including METAFONT, METAPOST, BibTeX, etc., not just TeX itself) and it is descriptive of a generic installation rather than a particular implementation (such as `emtex`).

The `texmf` root is not intended to be placed under an implementation-dependent directory. For example, `emtex/texmf` is wrong.

2.4 Top Level Directories

The top level directories in the `texmf` directory identify the major components of a TeX system. A site may omit any unneeded directories.

Although the TDS is designed to be implementation-independent, the TWG recognizes that some installers may wish to place other files in the TDS. For example, TeX administrators may wish to place all TeX-related files (including binaries, manual pages, pool files, and formats) in the `texmf` tree. This greatly simplifies administration of the TeX tree, especially if it is maintained by someone other than the system administrator. Moreover, it allows the administrator to construct a single "TeX server" (even in a heterogenous network environment).

The directories specified in this document are:

`tex` for input files used by TeX (see Section 3, "Macros").

`fonts` for font-related files (see Section 4, "Fonts").

`metafont` for (non-font) METAFONT files; (see Section 5, "Non-font METAFONT Files").

`metapost` for METAPOST files (see Section 6, "METAPOST").

`bibtex` for files used by BibTeX (see Section 7, "BibTeX").

`doc` for user documentation (see Section 8, "Documentation").

`program` for TeX applications. (In fact, the `tex`, `metafont`, `metapost`, and `bibtex` directories above may be seen as examples of this.) It may be convenient to store implementations (*emTeX*, *PCTeX*, etc.) as well as utilities (*dvips*, *MakeIndex*, etc.) in their own directories. Sites can store implementation-specific files (configuration files, compiled TeX format files, pool files, DLLs, etc.) and utility-specific files (configuration files, input files, etc.) in their own subdirectories.

`source` for sources. This includes both traditional program sources (for example, *Web2C* sources go in `texmf/source/web2c`) and LaTeX `dtx` sources (`texmf/source/latex`). The `dtx` files used for LaTeX distribution contain both the program sources and the documentation sources, so they are kept in the `source` tree.

`bin/`*`system`* for binaries. The *`system`* directories allow multiple implementations to share the common directory structure. For example, the binaries for a TeX system on a Sun workstation might be installed in `/texmf/bin/spsun413` (the name `spsun413` is one possible (ISO-9660 compliant) abbreviation for SPARC SunOS 4.1.3). Some TeX administrators may wish to put executables outside of `texmf` altogether.

The standard `texmf` tree provides no explicit location for locally-maintained files (e.g. letterheads), since that would require duplicating the entire tree. Consequently, sites are encouraged to maintain a separate tree for local styles and to use both trees in search paths. For example, `/usr/local/oratex` and `/usr/local/texmf`.

3 Macros

The common current practice of lumping files into a small number of directories has the disadvantage of making it difficult to determine which input files are associated with which macro packages, and may result in version mismatches and other errors when a package is upgraded, especially if the names of some files change.

To help solve this maintenance problem, the TDS specifies that macros shall be stored in separate directories, one for each package:

`texmf/tex/`*`format`*`/`*`package`*`/`

`format` is the name of the TeX format that uses these files. `plain`, `amstex`, `texinfo`, `latex`, and `musictex` are examples of *`format`*. By providing a format directory, path searching can be limited to only those directories that contain relevant files.

Although some of these formats can also be used as macro packages, the TDS nevertheless stores them as formats. By adjusting the TeX inputs search path, it is straightforward to use them as macro packages under another format, whereas placing them in the tree of another format would completely obscure their possible use as a format.

The following *`format`* names are reserved:

- `mft` is reserved for `mft` files. This exception follows standard practice.
- `generic` is reserved for input files that are useful across a wide range of formats. Generally, this means any format that uses the category codes of Plain TeX and does not rely on any particular format (e.g., `null.tex`).

The `plain` directory is for files which are useful with the Plain TeX format and others compatible with Plain: `testfont.tex`, `manmac.tex`, `webmac.tex`, etc., as well as `plain.tex` itself and the canonical example `story.tex`. The `generic` directory, by contrast, contains files which can also be used with LaTeX and other formats as well as Plain: `path.sty`, `texnames.sty`, `null.tex`, etc.

Thus, for almost every *format*, it is correct to search at least the *format* directory, and then the `generic` directory (in that order). Other directories may be searched as well. For example, the `amstex`, `plain`, and `generic` directories should be searched when using $\mathcal{AMS}$-TeX, because $\mathcal{AMS}$-TeX is compatible with Plain. This does not mean *only* those directories must be searched: a recursive search beginning at `texmf/tex` is a correct path for TeX inputs in a TDS tree.

package is the name of a package. `amslatex` and `babel` are examples of *package*.

The following *package* names are reserved:

- `base` is reserved for the base distribution for each format. Files used by `INITEX` when dumping format files shall also be stored in the `base` directory. For example, in the standard LaTeX distribution, the `ltx` files created during the build process shall be stored in the `base` directory along with the distribution.
- `misc` is reserved for packages that consist of a single file. It seems pointless to nest such a package/file within a directory of its own. It may also make subdirectory searching slower.
- `config` is reserved for configuration files. Configuration files are used by several formats including, for example, LaTeX and the Babel styles.
- `hyphen` is reserved for hyphenation patterns. These are typically used only by `INITEX`. This directory need exist only under the `generic` format directory.

In the case where a format consists of only a single file and has no auxiliary packages, that file can simply be placed in the *format* directory, instead of *format*/`base`. For example, Texinfo goes in `texmf/tex/texinfo/texinfo.tex`, not `texmf/tex/texinfo/base/texinfo.tex`.

4 Fonts

As with macros, fonts need to be stored in a hierarchical structure in order to make maintenance feasible. The TDS specifies the following (except for `PK` and `GF` files, which need additional structure, as detailed in the next sextion):

`texmf/fonts/`*type*`/`*supplier*`/`*typeface*`/`

type is the type of font file:

`source` font sources (METAFONT files, property lists, etc.).

`tfm` `TFM` files.

`vf` virtual fonts.

`afm` Adobe font metrics.

`pk` `PK` files.

`gf` `GF` files.

`type1` Type 1 fonts in `PFA` or `PFB` format or both.

`supplier` is the name of the font supplier or `public`. `adobe`, `monotype`, and `ams` are examples of `supplier`.

The `public` directory serves a practical purpose: it designates fonts which can be freely redistributed.

The name `ams` is precisely defined: the `ams` typeface directory contains *only* the fonts distributed by the AMS in the "amsfonts" collection.

`typeface` is the name of the typeface family. `cm`, `latex`, `euler`, `concrete`, `bookman`, and `courier` are examples of `typeface`.

The names `cm` and `latex` are precisely defined, as follows. The `cm` typeface directory contains only the 75 fonts defined in *Computers and Typesetting, Volume E.* The `latex` typeface directory contains only the fonts distributed with LaTeX in the base distribution.

For example:

```
texmf/fonts/tfm/public/cm/cmr10.tfm
texmf/fonts/type1/adobe/utopia/putr.pfa
```

For more information about font filenames, consult *Filenames for TeX fonts* (see Section C, "Related References" for a complete reference).

4.1 Font Bitmaps

The bitmaps for each font require special attention. In addition to the name of the font, two more characteristics are required to uniquely identify a bitmap font: the type of device (i.e., mode) for which the font was created and the resolution of the bitmap.

Following common practice, the TDS segregates fonts with different device types (modes) into separate directories. Consult `modes.mf` for recommended mode names. (See Section C, "Related References" for a complete reference.)

Two naming strategies are commonly used to identify the resolution of bitmap font files. On systems that allow long filenames, the convention is to include the resolution in the filename (in fact, this is how METAFONT itself names its output). For example, the 300 dpi bitmap of `cmr10` is named `cmr10.300pk`. On other systems, such as MS-DOS, the fonts are generally segregated into directories by resolution. For example, the same 300 dpi bitmap would be named `cmr10.pk` in a directory named `dpi300`.

Because the TDS cannot assume long filenames (see Section 2.2, "Constraints") it must use the latter scheme for naming fonts. Therefore, under the `pk` directory, the TDS specifies two more subdirectory levels:

`texmf/fonts/pk/mode/supplier/typeface/dpi/`

`mode` is the mode name which identifies the device type. This is usually the name of the METAFONT mode used to build the PK file. `ljfour`, `cx`, and `linotype` are examples of `mode`.

For fonts rendered as bitmaps by a program that does not distinguish between different output devices, the `mode` name shall be that of the program (e.g., `ps2pk`, `gsftopk`).

dpi specifies the resolution of the font. The directory name shall consist of the string `dpi` followed by the numeric value of the resolution in decimal. `dpi300`, `dpi329`, and `dpi1404` are examples of *dpi*.

`GF` files are stored in a parallel structure:

`texmf/fonts/gf/`*mode*`/`*supplier*`/`*typeface*`/`*dpi*`/`

4.1.1 Valid Font Bitmaps

The TWG recognizes that the use of short filenames has many disadvantages. The most vexing is that it results in the creation of dozens of different files with exactly the same name. At a typical site, `cmr10.pk` will be the filename for Computer Modern Roman 10 pt at 5–10 magnifications for 2–3 modes.

There was considerable discussion about how to minimize the disadvantages of this scheme. One result of that discussion was the decision to allow extensions to the basic naming scheme (such as `cmr10.300pk` or font library files), provided that the basic scheme is also supported. The following statement is the other result:

> In a TDS-conformant installation, all `PK` files *must* contain enough information to identify precisely how they were created; this includes mode, base resolution, and magnification used to create the font.

This information is easy to supply: a simple addition to the local modes used for building the fonts with METAFONT will automatically provide the required information. If you have been using a local modes file derived from (or that is, in fact) the complete `modes.mf` distribution from CTAN, the required information is already in your `PK` files. If not, a simple addition based on the code found in `modes.mf` (see Section C, "Related References" for a complete reference) can be made to your local modes file and the `PK` files rebuilt.

5 Non-font METAFONT Files

Most METAFONT input files are font programs or parts of font programs (see the previous section); however a few non-font input files do exist (`plain.mf`, `null.mf`, `expr.mf`, and `modes.mf`, for example).

The TDS specifies that these files shall be stored in:

`texmf/metafont/`*package*`/`

package is the name of a group of related non-font METAFONT files (for example, the `mfpic` macros).

The following *package* names are reserved:

- `base` is reserved for the standard METAFONT macros as described in *The METAFONTbook*, such as `plain.mf`, `expr.mf`, and the canonical example file `io.mf`.
- `misc` is reserved for METAFONT packages consisting of only a single file. One such example is `modes.mf`.

files are the names of individual METAFONT files like `plain.mf` or `null.mf`.

6 METAPOST

METAPOST is a picture-drawing language developed by John Hobby that is very much like Knuth's METAFONT, except that it outputs POSTSCRIPT instead of bitmaps.

The TDS specifies that the METAPOST input files and the support files for METAPOST-related utilities shall be stored in:

`texmf/metapost/`*`package`*`/`

package is the name of a group of related METAPOST files. At the moment there aren't any besides those in the standard distribution, but the TWG thought it wise to leave room for contributed packages that might be written in the future.

The following *package* names are reserved:

- `base` is reserved for the standard macros that come with METAPOST, such as `plain.mp`, `mfplain.mp`, `boxes.mp`, and `graph.mp`. This includes files used by `INIMP` when dumping mem files containing preloaded macro definitions.
- `misc` is reserved for METAPOST packages consisting of only a single file.
- `support` is reserved for special files required by the METAPOST utility programs, including a font map, a character adjustment table, and a subdirectory containing low-level METAPOST programs for rendering some special characters.

7 BIBTEX

BIBTEX databases and style files are commonly scattered in with other formats, making them hard to find and sometimes making BIBTEX hard to understand. The TDS specifies directories for them:

`texmf/bibtex/bib` contains BIBTEX database files.

`texmf/bibtex/bst` contains BIBTEX style files.

8 Documentation

Most packages come with some form of documentation. To make it easier for users to find this documentation, the TDS specifies that documentation shall be stored in a structure that parallels to some extent the `fonts` and `macros` directories.

There was much discussion about where documentation files should reside. One option was to place them in the same directory as the source files for the packages, but we felt it was important that users be able to find documentation in one place, not mixed in with the source files; most users have no interest in sources. Therefore, the TWG felt that a separate, but parallel, structure for documentation would (1) keep the documentation together and (2) make it as straightforward as possible for users to find the particular documentation they were after.

The TDS specifies that documentation files shall be organized as follows:

`texmf/doc/`*`category`*`/...`

`category` identifies the general topic of documentation that resides below it:

In the case of TeX formats, this will be the name of the format: `latex`, `plain`, `amstex`, etc. Within the category tree, the directory `base` is reserved for base documentation distributed by the format's authors.

Other categories include utility names, for example: `bibtex`, `makeindx`.

The following categories are reserved:

`help` is reserved for meta-information, such as FAQ's, David Jones' macro index, etc.

`general` is reserved for stand-alone documents not part of any package: Michael Doob's *A Gentle Introduction to TeX*, Joachim Schrod's *Components of TeX*, etc.

`man` is reserved for manual pages (Unix-style documentation).

`info` is reserved for processed Texinfo documents.

`html` is reserved for HTML documents.

The documentation directories may contain TeX sources, `DVI` files, PostScript files, text files, or whatever form of documentation will best explain the package.

A skeleton of the TDS directory tree under `texmf/doc`:

`help/`	meta-information
`  ctan/`	info about CTAN sites, `TeX-index`, etc.
`  faq/`	FAQs of `comp.text.tex`, etc.
`general/`	across programs, generalities
`  texcomp/`	*Components of TeX*
`  errata/`	`errata`, `errata[1-8]`
`format/`	name of a format (`latex` example follows)
`  base/`	for the base format
`  misc/`	for single-file package documentation
`  package/`	for *package* (`amslatex`, etc.)
`latex/`	
`  base/`	`ltnews*`, `*guide`, etc.
`  graphics/`	`grfguide`
`generic/`	for non-format specific TeX packages
`  german/`	`germdoc`
`  babel/`	
`program/`	TeX applications, by name (examples follow)
`tex/`	`texbook.tex`, `gentle`, etc.
`metafont/`	`mfbook.tex`, `metafont-for-beginners`, etc.
`metapost/`	`mpman`, `manfig`, etc.
`ams/`	
`  amsfonts/`	`amsfonts.faq`, `amfndoc`
`  amstex/`	`amsguide`, `joyerr`
`  amslatex/`	`amslatex.faq`, `amsldoc`
`web/`	`webman`, `cwebman`
`fontname/`	*Filenames for TeX fonts*
`fonts/`	
`  oldgerm/`	`corkpapr`
`  malvern/`	
`man/`	Unix-style manual pages
`html/`	HTML files
`info/`	GNU Info files, made from Texinfo sources

9 Package Distribution

As noted earlier, the goals of an archive and the goals of a working TeX system are sometimes in conflict. We encourage package authors to consider how their packages fit into a working TDS system and to document (or, preferably, automate) how the files should be moved from the archive structure into the TDS structure.

For example, a package for Martian language typesetting that works with Plain TeX and LaTeX might be distributed as the following:

```
martian-1.0/README
martian-1.0/doc/martdoc.tex
```

```
martian-1.0/doc/example.tex
martian-1.0/hyphen/marthyph.tex
martian-1.0/plain/martian.tex
martian-1.0/latex/martian.sty
martian-1.0/latex/OT1mart.fd
martian-1.0/fonts/martian.mf
martian-1.0/fonts/mart10.mf
martian-1.0/fonts/mart10.tfm
```

Documentation accompanying the distribution should describe how to construct the corresponding TDS structure:

```
texmf/doc/latex/martian/README
texmf/doc/latex/martian/martdoc.tex
texmf/doc/latex/martian/example.tex
texmf/doc/plain/martian/README
texmf/tex/generic/hyphen/marthyph.tex
texmf/tex/plain/martian/martian.tex
texmf/tex/latex/martian/martian.sty
texmf/tex/latex/martian/OT1mart.fd
texmf/fonts/source/public/martian/martian.mf
texmf/fonts/source/public/martian/mart10.mf
texmf/fonts/tfm/public/martian/mart10.tfm
```

Note that the main documentation is stored in the `latex` section of the documentation tree, but a `README` file also occurs in the `plain` section—presumably to point users to the `latex` files.

10 Summary

A skeleton of the TDS `texmf` directory tree:

`bibtex/`	BibTeX input files
`  bib/`	BibTeX databases of common interest
`  bst/`	BibTeX style files
`bin/*/`	binaries, by system type (optional)
`doc/`	see Section 8, "Documentation"
`fonts/`	font-related files
`  `*`type`*`/`	file type (e.g., `PK`)
`    `*`mode`*`/`	type of output device (for `PK` and `GF` only)
`      `*`supplier`*`/`	name of a font supplier (e.g., `public`)
`        `*`typeface`*`/`	name of a typeface (e.g., `cm`)
`          `*`dpi`*`/`	font resolution (for `PK` and `GF` only)
`metafont/`	METAFONT (non-font) input files
`  base/`	base distribution (e.g., `plain.mf`)
`  misc/`	single-file packages (e.g., `modes.mf`)
`  `*`package`*`/`	name of a package (e.g., `mfpic`)
`metapost/`	METAPOST input and support files
`  base/`	base distribution (e.g., `plain.mp`)
`  misc/`	single-file packages
`  `*`package`*`/`	name of a package
`  support/`	support files for METAPOST-related utilities
`program``/`	TeX applications, by name (e.g., `makeindx` and `dvips`)
`source/`	program source code by name (e.g., `web2c`, `latex`)
`tex/`	TeX input files
`  `*`format`*`/`	name of a format (e.g., `plain`)
`    base/`	base distribution for format
`    config/`	configuration files for format (e.g., `texsys.cfg`).
`    misc/`	single-file packages
`    `*`package`*`/`	name of a package (e.g., `graphics`)
`  generic/`	format-independent packages
`    hyphen/`	hyphenation patterns
`    misc/`	single-file format-independent packages (e.g. `texnames.sty`).
`    `*`package`*`/`	name of a package (e.g., `babel`)

A Implementation Issues

We believe that the TDS has the potential to bring a great deal of order to the current anarchic state of many TeX installations. In addition, by providing a common frame of reference, it will ease the burden of documenting administrative tasks. Finally, it is a necessary part of any reasonable system of true "drop-in" distribution packages for TeX.

A.1 Adoption of the TDS

We recognize that adoption of TDS will not be immediate or universal. Most TeX administrators will not be inclined to make the switch until:

- Clear and demonstrable benefits can be shown for the TDS.
- TDS-compliant versions of all key programs are available in ported, well-tested forms.
- A "settling" period has taken place, to flush out problems. The public release of this document is the first step in this process.

Consequently, most of the first trials of the TDS will be made by members of the TDS committee and/or developers of TeX-related software. Indeed, some of this has taken place during the course of our deliberations (see Section C, "Related References" for a sample tree available electronically). These trials may result in minor modifications to the TDS specification. They will certainly result in the production of a substantial number of TDS-compliant packages.

Once installable forms of key TDS-compliant packages are more widespread, some TeX administrators will set up TDS-compliant trees, possibly in parallel to existing production directories. This testing will flush out problems that were not obvious in the confined settings of the developers' sites. In particular, it should help to flush out OS and package dependencies, package interdependencies, and other esoteric details that were not considered by the committee.

Finally, after most of the dust has settled, even conservative TeX administrators will begin to adopt the TDS. Eventually, most TeX sites will have adopted the new structure, and most common packages will be readily available in TDS-compliant form.

A.2 Expediting the Process

We believe that the process described above will occur fairly quickly. The TDS committee spans a wide range of interests in the TeX community. Consequently, we believe that most of the key issues involved in defining a workable TDS definition have been covered, often in detail.

Further, a number of TeX developers have been consulted about implementation issues, and have been trying out suggested TDS formats. Thus, we hope for few surprises as implementation gets under way.

Finally, there are several (current or prospective) publishers of TeX-based CD-ROMs. These publishers are highly motivated to work out details of TDS implementation, and their products will provide inexpensive and convenient ways for experimentally-minded TeX administrators to try out the TDS.

Efforts are under way to set up a "TDS Registry" that will coordinate the assignment of TDS-compliant directory names and provide a definitive database of TDS-compliant software distributions.

A.3 Recursive Subdirectory Searching

Recursive subdirectory searching is the ability to specify that TEX (or any program in the TEX system) should search not only a specified directory, but recursively through all directories below that directory.

This feature is already supported by many implementations of TEX, including, but not limited to, *emTEX* (and its drivers), *PubliC TEX*, *Web2C*, *Y&YTEX*, *dvips(k)*, *xdvi(k)*, and DECUS TEX for VMS.

Even if your TEX implementation does not directly support subdirectory searching, you may find it useful to adopt the structure if you do not use many fonts or packages. For instance, if you only use Computer Modern and AMS fonts, you can store them in the TDS layout and list the directories individually in configuration files or environment variables.

The TWG recognizes that subdirectory searching places an extra burden on the system and may be the source of performance bottlenecks, particularly on slower machines. Nevertheless, we feel that subdirectory searching is imperative to a well-organized TDS, for the reasons stated above. Implementors are encouraged to provide extensions to the basic principle of subdirectory searching to avoid performance problems. One possible extension is the use of a "database" of filenames (this can be as simple as a recursive directory listing) that is consulted before directory searching begins. If a match is found in the database, subdirectory searching is not required, and performance is thus independent of the number of subdirectories present on the system.

A.4 Subdirectory Searching Syntax

Different implementations have elected to use different syntaxes for specifying subdirectory searching. In the interest of typographic clarity, the examples here do not use the `replaceable` font.

dvips: via a separate `TEXFONTS_SUBDIR` environment variable.

emTEX: `t:\directory!!`; `t:\directory!` for a single level of searching.

Kpathsea: `texmf/subdir//`

VMS: `texmf:[directory...]`

xdvi (patchlevel 20): `texmf/subdir/**`; `texmf/subdir/*` for a single level of searching.

Y&Y TEX: `t:/directory//` or `t:\directory\\`.

A.5 Mixed Case Filenames

One place where mixed-case names are known to occur is in LATEX font descriptor files. Fortunately, LATEX does not rely on case alone to distinguish between these files. MS-DOS, OS/2, Windows NT, and MacOS file systems are not case-sensitive; consequently a request for `OT1cmr.fd` can always be satisfied by `ot1cmr.fd` (or `OT1CMR.FD`). Unix platforms, which do have case-sensitive file systems, can use links (which most systems now support) to overcome any restrictions that are encountered. At installation time, a script may produce a whole TDS tree with long case-sensitive names that consist only of symbolic links to the CD-ROM file system. The space overhead produced by this "link farm" (at 1 KB per link) may be negligible compared to the space required for all the files.

B Is There a Better Way?

Defining the TDS required many compromises. Both the overall structure and the names of the individual directories were arrived at by finding common ground among many opinions. The driving forces were achievability (in terms of what could technically be done and what could reasonably be expected from developers) and regularity (files grouped together in an arrangement that "made sense").

The arrangement actually adopted tends to spread files out into two or three places (macros, documentation, and fonts, for example, are spread into different sections of the tree right at the top level).

B.1 A Better Macro Structure

The TWG settled on the `format/package` arrangement after long discussion about how best to arrange the files.

The primary alternative to this arrangement was a scheme which reversed the order of these directories: `package/format`. This reversed arrangement has a strong appeal: it keeps all of the files related to a particular package in a single place.

In the end, the `format/package` structure won for a couple of reasons:

- It is closer in spirit to the arrangements that are actually in use already. The fact that several members of the TWG already have arrangements that are similar to the TDS demonstrates that this arrangement is usable (proof by example). The alternative arrangement is not in use at any known site, and the TWG felt it wrong to mandate something with which there is no practical experience.
- The alternative arrangement spreads out the files needed by the system. The files that must be found using subdirectory searching may be spread out in a wide, shallow tree. This spreading effect could have a profound impact on the efficiency of subdirectory searching.

B.2 A Better Font Structure

The TDS mandates the use of `dpi300/cmr10.pk` for font naming syntax as it is the only portable form across present-day operating systems. Extensions to this recommendation, such as `cmr10.300pk` or the use of font library files, may optionally be employed on those systems that support this filename syntax, as long as the portable form of the filename is also supported.

The TWG struggled more with the font directory structure than anything else. The alternative arrangement we considered was:

`texmf/fonts/supplier/typeface/type/`

In fact, many sites already use this alternative arrangement. (It is the arrangement suggested by the *Web2C* distribution.)

This arrangement greatly improves the maintainability of the font tree, but, unless all the programs that search this tree employ some form of caching, there are serious performance concerns. For example, in order to find `cmr10.tfm`, TeX would potentially have to search through all the directories that contain `pk` files in all modes and at all resolutions. Clearly, this could make TeX unusable unless it had some method for "short circuiting" the search.

In the end, a straw poll of developers revealed considerable resistance to implementing sufficient caching mechanisms, so this other arrangement was abandoned. The arrangement settled on for the TDS allows the search tree to be restricted to the correct type of file, at least. Concerns about efficiency remain, but there seems to be no more we can do.

B.2.1 Where Does the Mode Go?

Another discussion arose in the effort to determine where the mode should go. The two alternative contenders were

`texmf/fonts/pk/`*`supplier/typeface/mode/dpi/`*

and

`texmf/fonts/pk/`*`mode/dpi/supplier/typeface/`*

In the former case, it isn't easy to limit subdirectory searching to only the mode required for the particular printer you are using. And in the latter case, leaving `dpi`*`NNN`* at the bottom allows that level to be omitted on systems which allow long filenames (e.g., `cmr10.300pk`).

C Related References

This appendix is for pointers to related files and other documents.

All discussions were held by email. The discussions are archived on `shsu.edu:[twg.tds]`.

In this document, "CTAN:" means the root of a anonymous ftp CTAN tree. This is both a host name and a directory name. For example:

```
ftp.dante.de:/tex-archive
ftp.shsu.edu:/tex-archive
ftp.tex.ac.uk:/tex-archive
```

Finger `ctan@ftp.shsu.edu` for a complete list of CTAN sites.

Filenames for TeX fonts is availble from CTAN:`documentation/fontname`.

A complete set of METAFONT modes is available from CTAN:`fonts/modes/modes.mf`. This file includes eight-character-or-fewer recommended mode names.

Components of TeX by Joachim Schrod is available from CTAN:`documentation/components-of-TeX`.

A sample tree is available from CTAN:`tds`.

`ftp.math.utah.edu:pub/tex/bib` has a huge BibTeX collection, both databases and styles.

D About the Committee

The TWG had no formal meetings (many, but not all, members attended TUG '94).

The following individuals participated in the discussions that led to this draft:

Beeton, Barbara (`bnb@math.ams.org`). Composition systems specialist and member of the technical support staff, Electronic Products and Services Department, American Mathematical Society, Providence, RI, U.S.A. Editor, *TUGboat*; charter member and Board member, TEX Users Group.

Berry, Karl (`kb@cs.umb.edu`). Maintainer of *Web2C*, Eplain, `modes.mf`, et al.

Brown, Vicki (`vlb@cfcl.com`). Editor of "Prime Time TEXcetera" from Prime Time Freeware. Vicki has been doing computer-aided document preparation for more than a decade. In her spare time, she works as a Quality Engineer for Apple.

Carlisle, David (`carlisle@ma.man.ac.uk`). Research Associate, Mathematics Department, Manchester University, UK. Member of the LATEX project team.

Jeffrey, Alan (`alanje@cogs.susx.ac.uk`). Lecturer in Computer Science and Artificial Intelligence at the School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK. Member of the LATEX project team.

MacKay, Pierre (`mackay@cs.washington.edu`). Professor, Dept. of Classics and Dept. of Near Eastern Languages, University of Washington. Together with Richard Furuta took over as Unix Site Coordinator for the distribution of TEX from Robert Morris, too many years ago to be remembered. Continues to support UnixTEX and the production of beautiful books, particularly in the humanities.

Morin, Rich (`rdm@cfcl.com`). President of Prime Time Freeware, publishers of "Prime Time TEXcetera." Rich has been programming computers for more than twenty-five years and has written more than 100 columns in magazines such as *SunExpert* and *UNIX Review*.

Rahtz, Sebastian (`sebastian.rahtz@cl.cam.ac.uk`). Production Methods Analyst, Elsevier Science Ltd. (TEX Users Group Technical Council liaison for the working group.)

Schrod, Joachim (`schrod@iti.informatik.th-darmstadt.de`). Scientific Assistant, Computer Science Department, Technical University of Darmstadt, Germany. Works on formal descriptions of user interface tools and author systems. Active TEXie since 1981; Secretary of DVI Driver Standards Committee (1991–1994), member of several other TUG TWGs, a founding member of DANTE e.V.

Tachikawa, Elizabeth (`unixtex@u.washington.edu`). Formerly associate director, Northwest Computing Support Center, at the University of Washington, and the organizing genius behind much of the best part of UnixTEX. Some of the original inspiration for the TDS comes from Elizabeth's early work to rationalize UnixTEX. Continues to contribute advice and support to UnixTEX.

Vieth, Ulrik (`vieth@thphy.uni-duesseldorf.de`). Graduate student in theoretical physics, maintainer of a local TEX installation for a few months, adapted METAPOST to work with *Web2C* in early 1995.

Walsh, Norman (`norm@ora.com`). Production tools specialist, O'Reilly & Associates, Inc. 90 Sherman Street, Cambridge, MA 02140. Author of *Making TEX Work*. (TDS Committee Chair.)