

# *Cahiers* **GUT** *enberg*

## ☞ TDS : UNE STRUCTURE DE RÉPERTOIRES POUR LES FICHIERS TEX

*Cahiers GUTenberg*, n° 44-45 (2004), p. 83-114.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_2004\\_\\_44-45\\_83\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_2004__44-45_83_0)>

© Association GUTenberg, 2004, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.



---

# TDS : une structure de répertoires pour les fichiers T<sub>E</sub>X

---

Groupe de travail TWG-TDS\*

*Traduction française de Jean-Côme Charpentier<sup>†</sup>*  
*Postface de Fabrice Popineau*

**Avertissement.** Cette note est la traduction française du document *A Directory Structure for T<sub>E</sub>X Files* publié par le *TUG Working Group on a T<sub>E</sub>X Directory Structure* et qui est disponible, en divers formats, à <http://tug.org/tds/>. Elle n'a aucun caractère officiel. Cette traduction est publiée ici avec l'aimable autorisation de Karl Berry au nom du TWG-TDS. La mise en page de cette traduction suit celle initiale de la classe `tdsguide.cls`, mais adaptée à la maquette des *Cahiers GUTenberg*. [N.D.L.R.]

*Copyright © 1994, 1995, 1996, 1997, 1998, 1999, 2003, 2004 T<sub>E</sub>X Users Group. Permission to use, copy, and distribute this document without modification for any purpose and without fee is hereby granted, provided that this notice appears in all copies. It is provided "as is" without expressed or implied warranty. Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the modifications are clearly marked and the document is not represented as the official one. This document is available on any CTAN host (see Appendix D). Please send questions or suggestions by email to [tds@tug.org](mailto:tds@tug.org). We welcome all comments. This is version 1.1.*

---

## Traduction française **non officielle**

---

Il est permis d'utiliser, de copier et de distribuer ce document *sans modification* dans n'importe quel but et sans frais à condition que cette note (anglaise) apparaisse dans toutes les copies. Il est fourni tel quel sans garantie explicite ou implicite. Il est permis de copier et de distribuer des versions modifiées de ce document sous les conditions de la copie verbatim, étant entendu que ces modifications seront clairement indiquées et que ce document ne sera pas présenté comme la version officielle. Ce document est disponible sur tous les sites CTAN (voir annexe D). Merci d'envoyer questions ou suggestions à [tds@tug.org](mailto:tds@tug.org). Nous accueillerons avec plaisir tous les commentaires. Il s'agit ici de la version 1.1.

---

\* TWG-TDS = *TUG Working Group on a T<sub>E</sub>X Directory Structure*. Sur la composition de ce groupe, voir la section E.

<sup>†</sup> La présente traduction est basée sur une celle de la version datant de 1999 (v 0.9996) qui avait été traduite conjointement avec Vincent Vaquin.

---

## Avant-propos du traducteur

Lorsqu'on découvre T<sub>E</sub>X, un des aspects déroutants de ce système est le nombre impressionnant de fichiers installés sur son disque dur. Il y en a tant que l'on ne sait plus où donner de la tête. Le problème devient critique lorsqu'on souhaite rapatrier une fonte, un fichier de style, une classe, etc., depuis un site abbrctan (par exemple) sur son propre système.

Nombre de messages fleurissent sur la liste de diffusion de GUTenberg ou les forums de discussion demandant « Bon, j'ai récupéré toto.mp (ou toto.sty ou toto.cls ou toto.pdf) mais je le mets où ? »

Dans le « je les mets où ? » on a en fait deux questions. La première sous question est celle que tout le monde se pose très rapidement : « je le mets où pour que T<sub>E</sub>X arrête de me dire qu'il ne trouve pas le fichier ? ». Force est de constater que des pseudo-conseils sont trop souvent prodigués indiquant de le placer en un endroit où se situe déjà un autre fichier .mp (respectivement .sty, .cls, .pdf). Certes, cela va fonctionner techniquement mais on se prépare alors des lendemains douloureux. La seconde sous question est moins évidente au départ mais elle devient cruellement évidente lors d'opérations particulières (migration sur un autre poste, problème de fichiers homonymes, etc.). On pourrait la formuler de la façon suivante : « je le mets où pour que mon système ne soit pas un bazar infernal ? ».

Pour donner une petite idée du problème, un système T<sub>E</sub>X courant (celui de l'auteur) va comporter de l'ordre d'un dizaine de milliers de fichiers t<sub>f</sub>m (métrique de fonte), de l'ordre de 5 000 fichiers f<sub>v</sub> (fonte virtuelle) et entre 500 et 1 000 fichiers sty. Le nombre de fichiers placés sous le répertoire racine principal de la distribution (par exemple /usr/share/texmf/ sous Unix) dépasse les 25 000. On voit vite que tout mettre dans un même répertoire n'est pas franchement une bonne idée !

Une première nécessité est donc d'organiser, si possible intelligemment, l'arborescence du système pour que les utilisateurs puissent s'y retrouver facilement. Ici, « utilisateur » est à prendre au sens large : ceux qui utilisent les fichiers sont à la fois les personnes tapant un document et des programmes (T<sub>E</sub>X lui-même mais aussi les programmes de production de fonte, de production d'index, de visualisation et d'impression, etc.). Une seconde nécessité a été d'inciter tout le monde à suivre la même organisation afin de rendre la tâche de certains programmes plus facile à gérer.

C'est pour ces raisons que le document *Une structure de répertoires pour les fichiers T<sub>E</sub>X* devient indispensable. Suivre les recommandations des auteurs de ce document assure de disposer d'une organisation de répertoires qui a fait l'objet d'une réflexion longue et attentive de personnes très compétentes et, d'autre part, va permettre à tous les programmes de fonctionner au mieux malgré les problèmes d'interdépendances.

---

Actuellement, *toutes* les distributions publiques ont adopté cette organisation des fichiers et les utilisateurs ont vraiment tout à gagner à y adhérer. Le document expliquant cette structure n'est vraiment pas long. De plus, sa lecture devient vraiment très rapide si l'on ne s'attarde pas trop sur l'exposé des raisons de tel ou tel choix effectué par le groupe de travail ayant conçu cette organisation. Enfin, le cadre proposé donne une certaine liberté à l'utilisateur et le document explique les choix possibles qui peuvent s'offrir. On peut noter en particulier le problème de savoir s'il faut ou non mettre tel ou tel fichier dans l'arborescence principale ou une arborescence locale. Ce document a donc en plus le mérite de faire réfléchir l'utilisateur sur des points qu'il aurait sans doute ignoré autrement.

Le dernier problème était que ce document est écrit en anglais et certains utilisateurs francophones pouvaient trouver là une excuse pour faire n'importe quoi sur leur système. Avec cette traduction, cette excuse n'est plus valable !

## 1 Introduction

$\TeX$  est un système de composition puissant et flexible utilisé par de nombreuses personnes dans le monde. Il est facilement portable et tourne quasiment sur tous les systèmes d'exploitation. Un effet de bord malheureux de la flexibilité de  $\TeX$  est qu'il n'y a pas une seule « bonne » façon de l'installer. Cela a donné lieu à de nombreux sites ayant des schémas d'installation différents. Le but premier de ce document est de décrire une structure de répertoires  $\TeX$  standard ( *$\TeX$  Directory Structure*, abrégé désormais en TDS) : une hiérarchie de répertoires pour les macros, les fontes et les autres fichiers dépendants de l'implantation du système  $\TeX$ . En ce qui concerne le côté pratique, ce document suggère également des façons d'ajouter le reste des fichiers dans une seule structure. La TDS a été pensée pour pouvoir fonctionner sur tous les systèmes modernes. En particulier, le groupe de travail technique (*Technical Working Group* : TWG) estime qu'elle est utilisable sous MacOS, MS-DOS, OS/2, Unix, VMS et Windows NT. Nous espérons que les administrateurs et les développeurs des distributions libres ou commerciales de  $\TeX$  adopteront ce standard. Ce document est prévu à la fois pour l'administrateur système d'un site et pour les personnes préparant des distributions  $\TeX$  – de tout de ce qui est nécessaire à un système complet fonctionnel jusqu'à une unique macro ou fichier de style. Il peut également aider les utilisateurs de  $\TeX$  à trouver leur chemin à l'intérieur des systèmes organisés de cette façon. Il ne s'agit pas d'un tutoriel : nous supposons nécessairement une connaissance de plusieurs parties d'un système  $\TeX$  fonctionnel. Si vous n'êtes pas familier avec les programmes et les fichiers auxquels nous nous référons, consultez les documents cités à l'annexe D.

## 1.1 Historique

La version 1.0 de la TDS est sortie en février 2003. La version 1.1 est sortie en juin 2004 avec les modifications suivantes :

- les sources des packages de T<sub>E</sub>X sont inclus dans `tex`, au lieu d’être placés à la racine de leurs propres sous-répertoires (section 3.1.1) ;
- nouveau répertoire de premier niveau `scripts` (section 3.6).
- nouveaux sous-répertoires `lig`, `opentype`, `truetype` et `type3` sous `fonts` (section 3.2) ;
- `enc`, `lig` et `map` utilisent tous les sous-répertoires  $\langle \text{syntaxe} \rangle / \langle \text{package} \rangle$  (section 3.2) ;
- les fichiers `pfm` doivent aller sous `type1` et les fichiers `inf` sous `afm` (section 3.2).

## 1.2 Rôle de la TDS

Le rôle de la TDS est de stabiliser l’organisation des packages logiciels du monde T<sub>E</sub>X qui sont installés et utilisés, éventuellement de façon simultanée, sur plusieurs plates-formes. À première vue, il peut sembler que le Comprehensive T<sub>E</sub>X Archive Network (CTAN) réalise au moins une partie de ce rôle mais, en réalité, ce n’est pas le cas. Le rôle du CTAN est de simplifier l’archivage et la distribution, pas l’installation ni l’utilisation. En réalité, les rôles de la TDS et du CTAN sont fréquemment en conflit comme nous le verrons. Pour une distribution, de nombreux types différents de fichiers doivent être combinés en une seule unité. Pour l’utilisation, il est habituel de répartir les fichiers (même des fichiers similaires) provenant d’un unique package dans des répertoires séparés, parfois même très éloignés.

## 1.3 Conventions

Dans ce document, « / » est utilisé pour séparer les composants des noms de fichiers comme, par exemple, `texmf/fonts`. Il s’agit de la convention Unix mais cette méthode n’est pas spécifique à Unix. Dans ce document, « T<sub>E</sub>X » signifie généralement le système T<sub>E</sub>X, y compris METAFONT, les pilotes DVI, les utilitaires, etc. et non pas seulement le programme T<sub>E</sub>X en lui-même. Le mot « package », dans ce document, a son sens habituel : un ensemble de fichiers distribués, installés et maintenus de façon unitaire. Ce *n’est pas* une extension L<sup>A</sup>T<sub>E</sub>X qui est un fichier de style complétant une classe de document. Nous utilisons les conventions typographiques suivantes :

**littéral** Un texte littéral tel que `nomfichier` est composé en fonte non proportionnelle.

***⟨variable⟩*** Un texte variable tel que  $\langle \text{package} \rangle$ , indiquant une classe de choses, est composé en italique à l’intérieur de crochets angulaires.

## 2 Généralités

Cette section décrit les propriétés communes à toute l'arborescence de la TDS.

### 2.1 Recherche de sous-répertoires

Les vieilles installations de  $\TeX$  stockaient un grand nombre de fichiers dans un seul répertoire, par exemple tous les fichiers TFM et/ou tous les fichiers sources  $\TeX$ . Cette organisation monolithique rendait plus difficile la maintenance d'un système  $\TeX$  : il était difficile de déterminer quels fichiers étaient utilisés par quels packages, quels fichiers nécessitaient d'être mis à jour lorsqu'une nouvelle version était installée ou quels fichiers devaient être supprimés lorsqu'un package était supprimé. C'était également une source d'erreur si deux packages ou plus avaient des fichiers sources ayant le même nom. Pour cette raison, le TWG a estimé que chaque package devait être dans un répertoire séparé. Nous reconnaissons cependant qu'une liste explicite de tous les répertoires pouvant être recherchés serait invivable. Un site peut souhaiter installer des dizaines de packages. De plus, une liste de tous les répertoires produirait des chemins de recherche de centaines de caractères de long, dépassant l'espace disponible sur certains systèmes. De même, si tous les répertoires étaient explicitement listés, l'installation ou la suppression d'un package impliquerait la modification d'un chemin ainsi que l'installation ou la suppression des fichiers. Cela serait très consommateur de temps et produirait facilement des erreurs, même avec les implantations qui fournissent un moyen de spécifier les répertoires en cours d'exécution d'une commande. Sur les systèmes ne disposant pas de cette possibilité de recherche en cours d'exécution, il faudrait recompiler le logiciel : ce serait une gêne intolérable. En définitive, le TWG a conclu qu'une TDS polyvalente nécessitait des implantations permettant une certaine forme de recherche implicite de sous-répertoires. Plus précisément, les implantations doivent permettre de spécifier que  $\TeX$ , METAFONT et leurs utilitaires satellites recherchent à la fois dans un répertoire donné et, récursivement, dans tous les sous-répertoires de ce répertoire lors d'une recherche d'un fichier. D'autres possibilités de recherche de sous-répertoires, par exemple des recherches récursives à un seul niveau, peuvent également être présentes. Nous recommandons aux concepteurs de distributions d'indiquer les sous-répertoires de recherches, en tant qu'options à l'installation ou en cours d'utilisation, pour tous les chemins. La TDS n'impose pas de syntaxe pour la spécification des recherches récursives mais nous recommandons aux concepteurs de distributions de penser au côté interactif (voir section B.2).

## 2.2 Enraciner l'arborescence

Dans ce document, nous indiquerons le répertoire racine de la TDS par « `texmf` » (pour « `TEX` et METAFONT »). Nous recommandons d'utiliser ce nom lorsque c'est possible mais le nom véritable de ce répertoire sera à la charge de celui qui installe. Sur les réseaux PC, par exemple, cela peut être mappé sur une spécification de disque logique telle que `T:`. De même, la localisation de ce répertoire sur le système sera dépendante du site. Elle peut être à la racine du système de fichiers. Sur les systèmes Unix, `/usr/local/share`, `/usr/local`, `/usr/local/lib` ou `/opt` sont des choix courants. Le nom `texmf` a été choisi pour plusieurs raisons : il reflète le fait que le répertoire contient des fichiers se rapportant à un système `TEX` entier (y compris METAFONT, METAPOST, BIB`TEX`, etc.) et pas seulement `TEX` lui-même ; il indique de façon descriptive qu'il s'agit d'une installation générique plutôt qu'une implantation particulière. Un site peut choisir d'avoir plus d'une hiérarchie TDS installée (par exemple, lors de l'installation d'une mise à jour). Cela est parfaitement légitime.

## 2.3 Ajouts locaux

La TDS ne peut pas spécifier précisément si un package est ou non un « ajout local ». Chaque site doit définir ce terme en fonction de ses propres conventions. Aux deux extrêmes, un site considérera « non local » tout fichier ne faisant pas partie de la distribution `TEX` installée alors qu'un autre site considérera « locaux » les seuls fichiers qui sont effectivement développés sur le site local et qui ne sont distribués nulle part ailleurs. Nous reconnaissons les deux méthodes courantes ci-dessous pour les ajouts locaux à une arborescence `texmf`. Les deux ont leur place et, en fait, certains sites les emploient de façon simultanée.

1. Une arborescence complètement séparée qui est donc elle-même une structure TDS. Par exemple `/usr/local/umbtex` à l'université du Massachusetts à Boston. C'est un autre exemple des hiérarchies `texmf` multiples mentionnées dans la section précédente.
2. Un répertoire nommé « `local` » à un niveau convenable. Par exemple, dans les répertoires `<format>`, `<package>` et `<fournisseur>` étudiés dans les sections suivantes. La TDS réserve le nom de répertoire `local` pour cet usage. Nous recommandons l'utilisation de `local` pour les fichiers de configuration adaptés au site tels que `language.dat` pour le package `Babel` ou `graphics.cfg` pour le package `graphics`. Les fichiers de configuration non modifiés d'un package devraient rester dans le répertoire de ce package. Le but est de séparer les fichiers créés ou modifiés localement des fichiers de la distribution afin de faciliter les nouvelles mises à jour.



Un tâche courante des ajouts locaux est la production dynamique de fichiers, par exemple les fontes PK par le script `mktexpk` (qui provient de `MakeTeXPK` dans `Dvips`). Un site peut stocker les fichiers produits directement au niveau de (au choix) :

- leur endroit standard dans l'arborescence TDS principale (si elle a été rendue globalement accessible en écriture) ;
- un autre endroit dans l'arborescence TDS principale (par exemple sous `texmf/fonts/tmp`) ;
- une arborescence TDS secondaire complète (comme précisé auparavant) ;
- un autre répertoire convenable (peut-être sous `/var`, par exemple `/var/spool/fonts`).

Il n'y a pas de solution unique adaptée à tous les sites.

## 2.4 Nom de fichiers dupliqués

Plusieurs fichiers ayant le même nom peuvent exister dans une arborescence TDS. La TDS ne spécifie généralement pas lequel des deux fichiers ayant le même nom dans un chemin de recherche sera trouvé ; par conséquent, généralement, la seule façon d'être certain de trouver un fichier donné est d'avoir un nom unique. Cependant, la TDS exige que les implantations acceptent les exceptions suivantes :

- Les noms des fichiers sources  $\TeX$  doivent être uniques au niveau de chaque sous-répertoire de premier niveau de `texmf/tex` et `texmf/tex/generic` mais pas sur toute l'arborescence de `texmf/tex`. C'est-à-dire que des formats  $\TeX$  différents peuvent avoir des fichiers ayant le même nom (la section 3.1 étudiera tout cela plus en détails). De cette façon, il n'y aura pas besoin de spécification de chemins dépendants du format. Par exemple, une recherche récursive débutant sur `texmf/tex` n'aura pas à spécifier d'autres répertoires. Ainsi, les implantations doivent fournir des spécifications de chemins dépendants du format, par exemple via des scripts ou des fichiers de configuration.
- De nombreuses fontes auront le même nom (par exemple `cmr10.pk`) ; voir la section 3.2.2. Les implantations doivent distinguer ces fichiers en regardant le mode et la résolution.

Toutes les implantations que nous connaissons ont ces propriétés. Les endroits où des noms dupliqués sont fréquemment rencontrés ne sont pas exceptionnels.

- Les noms des fichiers sources METAFONT (contrairement à ceux des fontes bitmap) doivent être uniques à l'intérieur de `texmf/fonts`. En pratique, il y a des problèmes avec certaines variantes de *Computer Modern* qui contiennent des fichiers légèrement modifiés appelés `punct.mf`, `roman1.mf`, et ainsi de suite. Nous croyons que la seule solution valide est de renommer les fichiers dérivés afin de les rendre uniques.

### 3 Répertoires de premier niveau

Les répertoires sous la racine `texmf` indiquent les composants majeurs d'un système  $\TeX$  (voir section 4 pour un récapitulatif). Un site peut omettre tout sous-répertoire non nécessaire. Bien que la TDS, par nature, puisse spécifier des endroits précis seulement pour des fichiers dépendants de l'implantation, nous reconnaissons que ceux qui installent peuvent souhaiter placer d'autres fichiers sous `texmf` afin de simplifier l'administration d'une arborescence  $\TeX$ , en particulier si elle est maintenue par quelqu'un d'autre que l'administrateur système. Par conséquent, des répertoires de premier niveau supplémentaires peuvent être créés. Les répertoires de premier niveau indiqués par la TDS sont :

`tex` pour les fichiers  $\TeX$  (section 3.1).

`fonts` pour les fichiers concernant les fontes (section 3.2).

`metafont` pour les fichiers METAFONT qui ne sont pas des fontes (section 3.3).

`metapost` pour les fichiers METAPOST (section 3.4).

`bibtex` pour les fichiers BIB $\TeX$  (section 3.5).

`scripts` pour les exécutable dépendants de la plate-forme (section 3.6).

`doc` pour la documentation de l'utilisateur (section 3.7).

`source` pour les sources. Cela comprend à la fois les programmes sources traditionnels (par exemple, les sources de Web2C vont dans `texmf/source/web2c`) et, par exemple, les fichiers sources  $\LaTeX$  `.dtx` qui vont dans `texmf/source/latex`. La TDS ne spécifie aucune structure pour le répertoire qui est sous `source`. `source` est destiné aux fichiers qui ne sont pas nécessaires à l'exécution de n'importe quel programme  $\TeX$ . Il ne doit être inclus dans aucun chemin de recherche. Par exemple, `plain.tex` ne doit pas se trouver sous `texmf/source`, même s'il s'agit bien d'un « fichier source » dans le sens qu'il n'est pas produit à partir d'un autre fichier (il va dans `texmf/tex/plain/base` comme expliqué à la section 3.1).

***<implantation>*** pour les implantations (par exemple `emtex`, `vtex`, `web2c`), devant être utilisées pour n'importe quel but considéré comme adéquat par la personne ayant réalisé l'implantation ou par l'administrateur  $\TeX$ . En d'autres termes, les fichiers qui ne peuvent pas être partagés entre plusieurs implantations, tels que les fichiers de spool (`tex.pool`) et les fichiers de dump (`plain.fmt`) vont à ces endroits, en plus des fichiers de configuration de l'implantation. Voir section B.3 pour des exemples d'arbres *<implantation>* réels. De tels fichiers de configuration spécifiques à l'implantation ne devrait *pas* être trouvés en utilisant le chemin de recherche principal des sources  $\TeX$  (par exemple `TEXINPUTS`). Cela doit être réservé aux fichiers réellement lus par le compilateur  $\TeX$ . Voir section 3.1.1.

⟨*programme*⟩ pour les sources spécifiques au programme et les fichiers de configuration de n'importe quel programme satellite de T<sub>E</sub>X (par exemple `mft`, `dvips`). En réalité, les items `tex`, `metafont`, `metapost` et `bibtex` ci-dessus peuvent toutes être vus comme des instances de ce cas.

### 3.1 Macros

Les fichiers de macros T<sub>E</sub>X seront stockés dans des répertoires séparés, en fonction du format T<sub>E</sub>X et du nom de package (nous utilisons « format » avec son sens T<sub>E</sub>X traditionnel pour indiquer un ensemble précompilé de macros comme `latex`, `plain`, etc.) :

```
texmf/tex/⟨format⟩/⟨package⟩/
```

⟨*format*⟩ est un nom de format (par exemple : `amstex`, `latex`, `plain`, `texinfo`).

La TDS autorise le stockage à n'importe quel niveau des distributions pouvant être utilisées comme formats ou packages (par exemple, `Texinfo`, `Eplain`), à la discrétion de l'auteur du format ou de l'administrateur T<sub>E</sub>X. Nous recommandons que les packages utilisés en tant que formats sur un site donné soient stockés au niveau ⟨*format*⟩ : en réglant le chemin de recherche des sources T<sub>E</sub>X, on pourra directement les utiliser comme des packages de macros sous d'autres formats, alors que les placer dans une autre arborescence cachera totalement leur utilisation en tant que format. La TDS réserve les noms de ⟨*format*⟩ suivants :

- `generic`, pour les fichiers sources qui sont utiles pour un grand nombre de formats (par exemple `null.tex`, `path.sty`). Généralement, cela signifie tout format utilisant les codes de catégorie de Plain T<sub>E</sub>X et non relié à un format particulier. C'est donc différent des fichiers qui ne sont utilisés que sous Plain T<sub>E</sub>X (qui devront être placés sous `texmf/tex/plain`) comme, par exemple, `testfont.tex` et `plain.tex` lui-même.
- `local`, pour les ajouts locaux. Voir section 2.3.

Ainsi, pour presque tous les formats, il est nécessaire de chercher au minimum dans le répertoire ⟨*format*⟩ puis dans le répertoire `generic` (dans cet ordre). Les autres répertoires peuvent avoir besoin d'être examinés également, cela dépendant du format. Lors de l'utilisation d'*A<sub>M</sub>S*-T<sub>E</sub>X, par exemple, les répertoires `amstex`, `plain` et `generic` doivent être examinés, car *A<sub>M</sub>S*-T<sub>E</sub>X est compatible avec Plain.

⟨*package*⟩ est un nom de package T<sub>E</sub>X (par exemple : `babel`, `texdraw`).

Au cas où un format serait constitué d'un seul fichier et n'aurait aucun package auxiliaire, ce fichier peut être simplement placé dans le répertoire ⟨*format*⟩ au lieu de ⟨*format*⟩/base. Par exemple, `Texinfo` peut être placé dans `texmf/tex/`

`texinfo/texinfo.tex` et non dans `texmf/tex/texinfo/base/texinfo.tex`. La TDS réserve les noms de  $\langle package \rangle$  suivant :

- `base`, pour la distribution de base de chaque format, y compris les fichiers utilisés par  $\text{\TeX}$  lors du dumping des fichiers de format. Par exemple, dans la distribution  $\text{\TeX}$  standard, les fichiers `.ltx` créés au cours du processus de fabrication. Un autre exemple est celui des fichiers de pilote `.ini` pour les formats utilisés par la  $\text{\TeX}$  Live et d'autres distributions.
- `hyphen`, pour les motifs de césures, y compris le fichier de motifs américains `hyphen.tex` d'origine. Ceux-ci sont typiquement utilisés uniquement par  $\text{\TeX}$ . Dans la plupart des situations, ce répertoire n'a besoin d'exister que sous le répertoire `generic`.
- `images`, pour les fichiers sources d'images tels que les figures en PostScript encapsulé. Bien qu'il ne soit pas très intuitif de mettre ces fichiers dans un répertoire nommé « `tex` »,  $\text{\TeX}$  a besoin de lire ces fichiers pour obtenir la bounding box ou d'autres informations. Un mécanisme pour partager les sources des images entre  $\text{\TeX}$  et d'autres programmes de composition de texte (par exemple Interleaf, FrameMaker) est au-delà des prérogatives de la TDS. Dans la plupart des situations, ce répertoire n'a besoin d'exister que sous le format `generic`.
- `local`, pour les ajouts locaux et les fichiers de configuration. Voir section 2.3.
- `misc`, pour les packages constitués d'un seul fichier. Un administrateur ou le mainteneur de ce package peuvent créer des répertoires pour les packages à fichier unique à leur convenance au lieu d'utiliser `misc`.

### 3.1.1 Extensions

Le moteur  $\text{\TeX}$  a inspiré de nombreux développements visant à lui trouver des compagnons ou successeurs, on peut citer PDF $\text{\TeX}$ , Omega, etc. La TDS indique que les fichiers sources de ces programmes (utilisant une syntaxe « à la  $\text{\TeX}$  ») doivent être placés à l'intérieur du répertoire de premier niveau `tex`, soit au niveau supérieur, soit dans un sous-répertoire de format, même si le programme d'origine  $\text{\TeX}$  n'est pas capable de les lire. Par exemple :

```
texmf/tex/aleph
texmf/tex/enc tex
```

Il s'agit d'une modification par rapport à la TDS 1.0 qui spécifiait des répertoires de premier niveau  $\langle extension \rangle$  pour chacun de ces programmes. Nous considérons que cette nouvelle approche est préférable car :

- Les auteurs de packages sensibles au moteur permettent souvent à leur code de détecter le moteur en train d'être utilisé en émettant des messages d'erreur ou en s'adaptant aux circonstances de façon appropriée. De plus, lors de sa maturation,

un package peut s'adapter à un nombre croissant de moteurs. Ainsi, il est concevable qu'un package puisse être placé dans chacun des multiples répertoires de premier niveau à différents moments. Mettre tous ces packages sous le répertoire de premier niveau `tex` donne un endroit stable au cours du temps.

- Les utilisateurs ont besoin de pouvoir utiliser plusieurs moteurs et la configuration de différents chemins de recherche pour chaque moteur serait difficile et sujette à erreurs.

En pratique, avoir différents répertoires de premier niveau entraînait des difficultés pour toutes les personnes concernées – utilisateurs, auteurs de packages, administrateurs de site et distributeurs de systèmes. Prenez soin de comparer cette approche avec le sous-répertoire de premier niveau *(implantation)* (section 3) qui devait être utilisé pour les fichiers de configuration qui (sans doute) n'utilisaient pas la syntaxe  $\TeX$  et, en tous cas, ne devait pas être trouvé par le chemin de recherche principal des sources  $\TeX$ .

## 3.2 Fontes

Les fichiers de fontes sont stockés dans des répertoires séparés, classés par type de fichier puis (dans la plupart des cas) par fournisseur et par famille. Les fichiers PK et GF ont besoin d'une structure supplémentaire, ce que nous verrons en détail dans la prochaine section.

```
texmf/fonts/<type>/<fournisseur>/<famille>/
texmf/fonts/enc,lig,map/<sousrepertoire>/
```

*<type>* est le type de la fonte. La TDS réserve les noms de *<type>* suivant pour les types de fichiers habituels de  $\TeX$  :

- `afm`, pour les fichiers de métriques Adobe et les fichiers `inf` ;
- `gf`, pour les fichiers de fontes générique (generic font) bitmap.
- `opentype`, pour les fontes OpenType ;
- `pk`, pour les fichiers bitmap compressé (packed bitmap) ;
- `source`, pour les fichiers sources de fontes (fichiers METAFONT, listes de propriétés (property lists), etc.).
- `tfm`, pour les fichiers de métriques de fontes ( $\TeX$  font metric) ;
- `truetype`, pour les fontes TrueType ;
- `type1`, pour les fontes PostScript Type 1 (dans `pfa`, `pfb` ou tout autre format) et les fichiers `pfm` ;
- `type3`, pour les fontes PostScript Type 3 ;
- `vf`, pour les fontes virtuelles (virtual fonts).

La TDS réserve également les noms `enc`, `lig` et `map` pour, respectivement, le codage des fontes, les ligatures et les fichiers de `map`. Tous ces répertoires sont structurés de la même façon avec des sous-répertoires *<syntaxe>* puis

des sous-sous-répertoires (*package*). Chacun de ces types de fichiers sera recherché avec son propre chemin de recherche récursive. Comme d'habitude, les noms des fichiers effectifs doivent être uniques à l'intérieur de leur sous-arbre. Exemples :

```
fonts/map/dvipdfm/updmap/dvipdfm.map
fonts/map/dvips/lm/lm.map
fonts/enc/dvips/base/8r.enc
```

Les packages Fontname et Dvips ont plus d'exemples de types enc et map. Le programme afm2pl utilise des fichiers lig. Les fichiers pfm sont inclus dans le répertoire type1 au lieu d'être placés dans leurs propres répertoires pour deux raisons : 1) un fichier .pfm est toujours un supplément d'un fichier .pfb donné ; 2) ils doivent être installés depuis le même répertoire pour que des programmes Windows autres que T<sub>E</sub>X puissent les utiliser. Les fichiers inf sont placés dans le répertoire afm puisque les fichiers inf et afm peuvent être utilisés pour produire un pfm (malheureusement, Adobe Type Manager, et peut-être d'autres logiciels, demande que les fichiers pfb soient dans le même répertoire que les fichiers afm et inf pour leur installation). Comme d'habitude, un site peut omettre n'importe lequel de ces répertoires dans la mesure où ils ne servent pas. Le répertoire gf est un bon candidat pour une telle omission.

⟨*fournisseur*⟩ est un nom identifiant la source de la fonte (exemples : adobe, ams, public). La TDS réserve les noms de ⟨*fournisseur*⟩ suivant :

- ams, pour la bibliothèque de fontes *AMS*-fonds de l'American Mathematical Society.
- local, pour les ajouts locaux. Voir section 2.3.
- public, pour les fontes librement redistribuables où le fournisseur : (1) ne demande pas son propre répertoire (par exemple ams) ; (2) ne fabrique pas de fontes propriétaires (par exemple adobe). Il ne contient pas toutes les fontes librement distribuables en circulation et les fichiers placés à son niveau ne sont pas nécessairement strictement du domaine public.
- tmp, pour les fontes produites de façon dynamique comme c'est l'habitude sur certains systèmes. Il peut être omis s'il n'est pas utile.

⟨*famille*⟩ est le nom de la famille de fontes (exemples : cm, euler, times). La TDS réserve les noms de ⟨*famille*⟩ suivants :

- cm (au niveau de public), pour les 75 fontes définies dans *Computers and Typesetting, Volume E*.
- latex (au niveau de public), pour les fontes distribuées avec L<sup>A</sup>T<sub>E</sub>X dans une distribution de base.
- local, pour les ajouts locaux. Voir section 2.3.

Voici quelques exemples :

```
texmf/fonts/source/public/pandora/pnr10.mf
texmf/fonts/tfm/public/cm/cmr10.tfm
texmf/fonts/type1/adobe/utopia/putr.pfa
```

Pour une liste complète des noms de fournisseurs et de familles, on consultera *Filenames for T<sub>E</sub>X fonts* (voir annexe D).

### 3.2.1 Fontes bitmap

Les fichiers de fontes bitmap nécessitent deux caractérisations en plus de celles indiquées précédemment pour être identifiables de façon unique : (1) le type du périphérique (c'est-à-dire le mode) pour lequel la fonte a été créée ; (2) la résolution du bitmap. Suivant en cela une pratique courante, la TDS répartit les fontes sous différents répertoires en fonction des différents types de périphériques. Voir `modes.mf` à l'annexe D pour les recommandations des noms de modes. Certaines imprimantes ont plusieurs résolutions possibles (par exemple 300 dpi et 600 dpi) mais chaque résolution nécessitera un nom de mode différent. Rien de plus n'est nécessaire puisque, dans T<sub>E</sub>X, il est implicitement supposé qu'il n'y a qu'une unique résolution possible. Deux types de nomenclatures sont couramment utilisés pour identifier la résolution des fichiers de fontes bitmap. Sur les systèmes qui admettent les noms de fichiers longs (et pour le programme d'origine METAFONT lui-même), la résolution est incluse dans le nom de fichier (par exemple `cmr10.300pk`). Sur les systèmes qui ne supportent pas les noms de fichiers longs, les fontes sont généralement réparties en répertoires suivant leurs résolutions (par exemple `dpi300/cmr10.pk`). Comme la TDS ne peut pas exiger les noms de fichiers longs, nous devons utiliser le second schéma pour dénommer les fontes. Ainsi, nous aurons deux niveaux supplémentaires de sous-répertoires sous `pk` et `gf` :

```
texmf/fonts/pk/<mode>/<fournisseur>/<famille>/dpi<nnn>/
texmf/fonts/gf/<mode>/<fournisseur>/<famille>/dpi<nnn>/
```

`<mode>` est un nom par lequel on identifie le type de périphérique (par exemple : `cx`, `ljfour`, `modeless`). Normalement, il s'agit du nom du mode METAFONT utilisé pour construire les fichiers PK. Pour les fontes bitmap produites par un programme qui ne distinguerait pas ces différents types de périphériques, le `<mode>` sera simplement `modeless`. Le niveau `<mode>` ne doit pas être omis, même s'il n'y a qu'un seul mode qui sera effectivement utilisé.

`dpi<nnn>` spécifie la résolution de la fonte (par exemple : `dpi300`, `dpi329`). « dpi » signifie « dots per inch », c'est-à-dire pixels par pouce. Nous sommes conscient que dans de nombreux pays les pixels par millimètre sont utilisés mais l'unité dpi est trop ancrée dans le monde T<sub>E</sub>X pour que nous pensions en changer maintenant.

L'entier  $\langle nnn \rangle$  doit être calculé comme si l'on utilisait l'arithmétique et l'arrondi METAFONT, c'est-à-dire comme s'il s'agissait de l'entier utilisé par METAFONT dans son fichier de sortie `gf`. Nous reconnaissons cependant que de petites différences au niveau de la résolution sont une cause fréquente de frustration pour les utilisateurs et nous recommandons aux concepteurs de suivre le standard du pilote DVI niveau 0 (voir annexe D) pour la recherche de fontes bitmap en autorisant un écart de  $\pm 0,2\%$  (avec un minimum de 1) au niveau du  $\langle dpi \rangle$ .

Les implantations peuvent fournir des extensions à ce schéma de base pour les noms, telles que les noms de fichiers longs (comme dans le METAFONT d'origine) ou les fichiers de bibliothèque de fontes (comme dans les fichiers `.fli` de `emTEX`), dans la mesure où le schéma de base sera également supporté.

### 3.2.2 Noms de fontes bitmap valides

Le TWG a conscience que l'utilisation des noms de fichiers courts a de nombreux désavantages. Le plus vexant est que cela entraîne la création de dizaines de fichiers différents ayant le même nom. Sur un site typique, `cmr10.pk` sera le nom de fichier pour le Computer Modern Roman 10 pt sous 5 à 10 échelles différentes et pour 2 ou 3 modes différents (on se reportera à la section 2.4 qui étudiera de façon générale ce problème de duplication de noms de fichiers). Pour minimiser le problème, nous recommandons chaudement que les fichiers PK contiennent suffisamment d'informations pour identifier précisément la façon dont ils ont été créés : au moins le mode, la résolution de base et l'échelle utilisée pour créer la fonte. Cette information est facile à obtenir : un simple ajout dans les modes locaux utilisés pour la production des fontes avec METAFONT fournira automatiquement l'information requise. Si l'on a utilisé un fichier de modes locaux dérivé de (ou qui est simplement) `modes.mf` (voir annexe D), l'information requise est déjà dans les fichiers PK. Sinon, il suffit d'effectuer un simple ajout au niveau des fichiers de modes locaux, en se basant sur le code trouvé dans `modes.mf`, puis de reconstruire les fichiers PK.

### 3.3 Fichiers METAFONT hors fontes

La plupart des fichiers sources de METAFONT sont des programmes de fontes ou des parties de programmes de fontes et ils ont été couverts par la section précédente. Cependant, un petit nombre de fichiers n'étant pas des fichiers sources de fontes existent. Ces fichiers seront stockés dans :

```
texmf/metafont/ $\langle package \rangle$ /
```

$\langle package \rangle$  étant le nom d'un package METAFONT (par exemple, `mfpic`). La TDS réserve les noms de  $\langle package \rangle$  suivants :



- `base`, pour les fichiers de macros METAFONT de base, décrits dans *The METAFONTbook*, tels que `plain.mf` et `expr.mf`.
- `local`, pour les ajouts locaux. Voir section 2.3.
- `misc`, pour les packages METAFONT constitués seulement d'un fichier unique (par exemple, `modes.mf`). Un administrateur ou un mainteneur de package peuvent créer des répertoires pour des packages ayant un seul fichier à leur discrétion au lieu d'utiliser `misc`.

### 3.4 METAPOST

METAPOST est un langage de tracé graphique développé par John Hobby et qui est dérivé du METAFONT de Knuth. Son but principal est de produire du PostScript encapsulé au lieu de fontes bitmap. Les fichiers sources de METAPOST et les fichiers concernant les utilitaires satellites de METAPOST seront stockés dans :

```
texmf/metapost/⟨package⟩/
```

⟨*package*⟩ étant le nom d'un package METAPOST. À l'heure actuelle, aucun n'existe mais le TWG pense qu'il est prudent de laisser de la place pour les packages pouvant être écrits à l'avenir. La TDS réserve les noms de ⟨*package*⟩ suivants :

- `base`, pour les fichiers de macros METAPOST de base tels que `plain.mp`, `mfpain.mp`, `boxes.mp` et `graph.mp`. Cela inclut les fichiers utilisés par `iniMP` lors d'un dumping mémoire des fichiers contenant des définitions de macros pré-chargées.
- `local`, pour les ajouts locaux. Voir section 2.3.
- `misc`, pour les packages METAPOST constitués d'un seul fichier. Un administrateur ou un mainteneur de package peuvent créer des répertoires pour des packages ayant un seul fichier à leur discrétion au lieu d'utiliser `misc`.
- `support`, pour les fichiers sources supplémentaires exigés par les programmes satellites de METAPOST, ce qui comprend les fichiers `.map` des fontes, un tableau d'ajustement de caractères et un sous-répertoire contenant les programmes de bas niveau METAPOST pour le rendu de certains caractères spéciaux.

### 3.5 BIB $\TeX$

Les fichiers en rapport avec BIB $\TeX$  seront stockés dans :

```
texmf/bibtex/bib/⟨package⟩/
texmf/bibtex/bst/⟨package⟩/
```

Le répertoire `bib` est destiné aux fichiers de bases de données BIB $\TeX$  (`.bib`) et le répertoire `bst` aux fichiers de style (`.bst`). ⟨*package*⟩ est le nom d'un package BIB $\TeX$ . La TDS réserve les noms de ⟨*package*⟩ suivants (les mêmes noms sont réservés sous `bib` et `bst`) :

- `base`, pour les fichiers de bases de données et les styles standard  $\text{BIB}\TeX$  tels que `xampl.bib` et `plain.bst`.
- `local`, pour les ajouts locaux. Voir section 2.3.
- `misc`, pour les packages  $\text{BIB}\TeX$  constitués seulement d'un fichier unique. Un administrateur ou un mainteneur de package peuvent créer des répertoires pour des packages ayant un seul fichier à leur discrétion au lieu d'utiliser `misc`.

### 3.6 Scripts

Le répertoire de premier niveau `scripts` est destiné aux exécutable indépendants du système tels que Perl, Python ou les scripts shell ainsi que pour les fichiers de classe Java. Les sous-répertoires de `scripts` sont des noms de packages. Cela facilite la création des distributions en fournissant un endroit commun pour de tels programmes indépendants de la plate-forme. Le but n'est pas d'ajouter tous ces sous-répertoires à une commande de recherche de sous-répertoires d'un utilisateur, ce qui serait pratiquement impossible à gérer. Il s'agit plutôt d'y mettre des exécutable prévus initialement pour des scripts d'appel qu'une distribution pourra utiliser à partir de n'importe quel répertoire (et qui n'est pas spécifié par la TDS). De véritables scripts auxiliaires, plutôt que les scripts d'appel, qui seront exécutés directement par d'autres programmes peuvent également être placés à ce niveau. Cela étant, `scripts` sert également comme une plate-forme indépendante tout comme le répertoire standard `libexec` sous Unix. Nous recommandons l'utilisation d'extensions précisant le langage (telles que `.pl`, `.py`, `.sh`) pour ces fichiers, uniquement pour mieux identifier leurs noms. Comme le but de la TDS concernant le répertoire `scripts` est d'y placer des programmes ne devant pas être appelés directement par l'utilisateur, cela ne pose aucun problème. Par exemple, dans la distribution  $\text{T}\TeX$  Live, le programme `ConT $\text{E}$ Xt texexec`, exécutable directement par l'utilisateur, peut exister en tant que petit script d'appel dans chaque répertoire `bin/⟨plate-forme⟩/texexec` (qui sont en dehors de l'arborescence `texmf`), ce dernier ira tout simplement trouver et lancer `texmf/scripts/context/perl/texexec.pl`. Exemples :

```
scripts/context/perl/texexec.pl
scripts/context/ruby/example.rb
scripts/thumbpdf/thumbpdf.pl
```

La TDS ne spécifie pas d'emplacement pour les binaires exécutable dépendants de la plate-forme, qu'ils soient destinés directement ou non à l'utilisateur.

### 3.7 Documentation

La plupart des packages sont disponibles avec un certain type de documentation : manuel de l'utilisateur, fichiers d'exemples, guide du programmeur, etc. De plus,

beaucoup de fichiers indépendants ne faisant pas partie de packages ont été créés pour décrire des aspects variés du système T<sub>E</sub>X. La TDS indique que ces fichiers de documentation supplémentaires seront stockés dans une structure parallèle, jusqu'à une certaine profondeur, aux répertoires `fonts` et `tex` comme décrit ci-dessous :

```
texmf/doc/<catégorie>/...
```

*<catégorie>* indique le sujet général de la documentation qui est placé sous ce niveau. Par exemple, un nom de format T<sub>E</sub>X (`latex`), un nom de programme (`bibtex`, `tex`), un nom de langue (`french`, `german`), un format de fichier (`info`, `man`) ou d'autres composants du système (`web`, `fonts`). Une disposition possible est d'organiser `doc` par langue, avec tous les autres types de catégories au-dessous. Cela aidera les utilisateurs à trouver de la documentation dans la ou les langues avec lesquelles il se sentent plus à l'aise. Ni cette façon de faire ni aucune autre, bien sûr, n'est exigée. À l'intérieur de chaque arborescence *<catégorie>* d'un format T<sub>E</sub>X, le répertoire base est réservé pour la documentation de base distribuée par les mainteneurs du format. La TDS réserve les noms de catégories suivants :

- `general`, pour les documents indépendants, non spécifiques à un programme particulier (par exemple *Components of T<sub>E</sub>X* de Joachim Schrod).
- `help`, pour les méta-informations, telles que les FAQ's, le Catalogue T<sub>E</sub>X, etc.
- `info`, pour les documents traités par Texinfo. Les fichiers Info, comme n'importe quels autres, peuvent également être stockés en dehors de l'arborescence TDS avec une option lors de l'installation.
- `local`, pour les ajouts locaux. Voir section 2.3.

Le répertoire `doc` est prévu pour les fichiers de documentation indépendants de l'implantation et du système. Les fichiers dépendants de l'implantation devraient être stockés ailleurs, comme l'implantation ou l'administrateur T<sub>E</sub>X l'indique (par exemple, les fichiers d'aide sous VMS sont placés sous `texmf/vms/help`). Les répertoires de documentation peuvent contenir des sources T<sub>E</sub>X, des fichiers DVI, des fichiers PostScript, des fichiers textes, des exemples de fichiers sources ou n'importe quels autres formats utiles de documentation. Voir section 4.1 pour un récapitulatif.

## 4 Récapitulatif

Voici un squelette d'une arborescence TDS de répertoire `texmf`. Cela n'implique pas qu'il n'y ait que ces entrées de permises. Par exemple, un `local` peut être placé à n'importe quel niveau.

|                                    |                                                                               |
|------------------------------------|-------------------------------------------------------------------------------|
| <code>bibtex/</code>               | fichiers sources BIB <sub>T</sub> EX                                          |
| <code>bib/</code>                  | bases de données BIB <sub>T</sub> EX                                          |
| <code>base/</code>                 | distribution de base (ex : <code>xampl.bib</code> )                           |
| <code>misc/</code>                 | bases de données constituées d'un seul fichier                                |
| <code>&lt;package&gt;/</code>      | nom de package                                                                |
| <code>bst/</code>                  | fichiers de style BIB <sub>T</sub> EX                                         |
| <code>base/</code>                 | distribution de base (ex : <code>plain.bst</code> , <code>acm.bst</code> )    |
| <code>misc/</code>                 | styles constitués d'un seul fichier                                           |
| <code>&lt;package&gt;/</code>      | nom de package                                                                |
| <code>doc/</code>                  | voir section 3.7 et le récapitulatif ci-dessous                               |
| <code>fonts/</code>                | fichiers en rapport avec les fontes                                           |
| <code>&lt;type&gt;/</code>         | type de fichier (ex : <code>pk</code> )                                       |
| <code>&lt;mode&gt;/</code>         | type de périphérique de sortie (pour <code>pk</code> et <code>gf</code> )     |
| <code>&lt;fournisseur&gt;/</code>  | nom d'un fournisseur de fontes (ex : <code>public</code> )                    |
| <code>&lt;famille&gt;/</code>      | nom d'une famille (ex : <code>cm</code> )                                     |
| <code>dpi&lt;nnn&gt;/</code>       | résolution de fontes (seulement pour <code>pk</code> et <code>gf</code> )     |
| <code>&lt;implantation&gt;/</code> | implantations T <sub>E</sub> X par nom (ex : <code>emtex</code> )             |
| <code>local/</code>                | fichiers créés ou modifiés sur le site local                                  |
| <code>metafont/</code>             | fichiers sources METAFONT (hors fontes)                                       |
| <code>base/</code>                 | distribution de base (ex : <code>plain.mf</code> )                            |
| <code>misc/</code>                 | packages constitués d'un seul fichier (ex : <code>modes.mf</code> )           |
| <code>&lt;package&gt;/</code>      | nom d'un package (ex : <code>mfpic</code> )                                   |
| <code>metapost/</code>             | fichiers sources et satellite METAPOST                                        |
| <code>base/</code>                 | distribution de base (ex : <code>plain.mp</code> )                            |
| <code>misc/</code>                 | packages constitués d'un seul fichier                                         |
| <code>&lt;package&gt;/</code>      | nom d'un package                                                              |
| <code>support/</code>              | fichiers pour les utilitaires liés à METAPOST                                 |
| <code>mft/</code>                  | sources MFT (ex : <code>plain.mft</code> )                                    |
| <code>&lt;programme&gt;/</code>    | programmes satellites de T <sub>E</sub> X, par nom (ex : <code>dvips</code> ) |
| <code>source/</code>               | code source de programmes (ex : <code>latex</code> , <code>web2c</code> )     |
| <code>tex/</code>                  | fichiers sources T <sub>E</sub> X                                             |
| <code>&lt;moteur&gt;/</code>       | nom d'un moteur (ex : <code>aleph</code> ) ; éventuellement plus bas          |
| <code>&lt;format&gt;/</code>       | nom d'un format (ex : <code>plain</code> )                                    |
| <code>base/</code>                 | distribution de base d'un format (ex : <code>plain.tex</code> )               |
| <code>misc/</code>                 | packages constitués d'un seul fichier (ex : <code>webmac.tex</code> )         |
| <code>local/</code>                | ajouts locaux ou fichiers de config. pour un <code>&lt;format&gt;</code>      |

---

|                         |                                                        |
|-------------------------|--------------------------------------------------------|
| <i>&lt;package&gt;/</i> | nom d'un package (ex : graphics, mfnfss)               |
| generic/                | packages indépendants du format                        |
| hyphen/                 | motifs de césures (ex : hyphen.tex)                    |
| images/                 | fichiers sources graphique (ex : PostScript encapsulé) |
| misc/                   | packages indépendants du format à fichier unique       |
| <i>&lt;package&gt;/</i> | nom d'un package (ex : babel)                          |

#### 4.1 Récapitulatif de l'arborescence de la documentation

Voici un exemple de squelette d'une arborescence de répertoires TDS sous texmf/doc. Cela n'implique pas qu'il n'y ait que ces entrées de permises ni que cette structure doit suivre exactement ce qui est présenté ci-dessous. Comme indiqué précédemment, l'arborescence texmf/doc peut être organisée par langue. De cette façon, toute la documentation en français, disons, sera placée dans un sous-répertoire french. Dans ce cas, l'exemple de structure donnée ci-dessous sera celle placée au niveau du sous-répertoire donné d'une langue.

|                         |                                                         |
|-------------------------|---------------------------------------------------------|
| ams/                    |                                                         |
| amsfonts/               | amsfonts.faq, amfndoc                                   |
| amslatex/               | amslatex.faq, amslldoc                                  |
| amstex/                 | amsguide, joyerr                                        |
| bibtex/                 | BIBT <sub>E</sub> X                                     |
| base/                   | btxdcoc.tex                                             |
| fonts/                  |                                                         |
| fontname/               | <i>Filenames for T<sub>E</sub>X fonts</i>               |
| oldgerm/                | corkpapr                                                |
| <i>&lt;format&gt;/</i>  | nom d'un format T <sub>E</sub> X (ex : generic, latex)  |
| base/                   | pour la distribution de base                            |
| misc/                   | pour les contributions constituées d'un seul fichier    |
| <i>&lt;package&gt;/</i> | pour <i>package</i>                                     |
| general/                | tour des programmes, généralités                        |
| errata/                 | errata, errata[1-8]                                     |
| texcomp/                | <i>Components of T<sub>E</sub>X</i>                     |
| help/                   | méta-information                                        |
| ctan/                   | informations sur les sites miroirs CTAN                 |
| faq/                    | FAQs de comp.text.tex, etc.                             |
| info/                   | fichiers GNU Info, réalisés à partir de sources Texinfo |
| latex/                  | exemple de <i>&lt;format&gt;</i>                        |
| base/                   | ltnews*, *guide, etc.                                   |
| graphics/               | grfguide                                                |
| local/                  | documentation spécifique au site                        |
| man/                    | pages de man Unix                                       |

---

|                                 |                                                                                     |
|---------------------------------|-------------------------------------------------------------------------------------|
| <code>&lt;programme&gt;/</code> | programmes du monde $\TeX$ , classés par nom                                        |
| <code>metafont/</code>          | <code>mfbook.tex</code> , <code>metafont-for-beginners</code> , etc.                |
| <code>metapost/</code>          | <code>mpman</code> , <code>manfig</code> , etc.                                     |
| <code>tex/</code>               | <code>texbook.tex</code> , <i>A Gentle Introduction to <math>\TeX</math></i> , etc. |
| <code>web/</code>               | <code>webman</code> , <code>cwebman</code>                                          |

## A Points non spécifiés

La TDS ne peut pas s'occuper des aspects suivants d'un système  $\TeX$  fonctionnel :

1. L'emplacement des programmes exécutables : c'est trop dépendant du site pour pouvoir recommander un endroit, encore moins l'exiger. Un site peut placer les exécutables en dehors de l'arborescence `texmf` (par exemple dans `/usr/local/bin`), au niveau d'un répertoire dépendant de la plate-forme à l'intérieur de `texmf` ou n'importe où ailleurs.
2. Les packages mis à jour lorsque de nouvelles versions sont réalisées : nous ne pouvons trouver aucune façon d'introduire des spécifications de versions dans `texmf` qui feraient plus de bien que de mal ou qui seraient adaptées à la plupart des installations.
3. L'emplacement des fichiers spécifiques à l'implantation (par exemple les fichiers  $\TeX$  .`fmt`) : par nature, ils doivent être laissés à la discrétion de ceux qui implantent ou maintiennent  $\TeX$ . Voir section B.3.
4. Ce qui doit être précisément considéré comme « local » et où de tels fichiers locaux doivent être installés. Voir section 2.3 pour plus de détails.

### A.1 Noms de fichiers portables

La TDS ne peut pas exiger une quelconque restriction sur les noms de fichiers dans l'arborescence puisque les noms de beaucoup de fichiers  $\TeX$  existants ne se conforment à aucun schéma standard. Pour le bien des personnes voulant réaliser une distribution ou une installation  $\TeX$  portable, cependant, nous soulignons ici certaines restrictions nécessaires. Les spécifications de la TDS elles-mêmes sont compatibles avec celles-ci. La norme ISO-9660 est le seul système de format de fichiers universellement accepté pour les CD-ROM. Un sous-ensemble des limitations indiquées est partagé par tous les systèmes d'exploitation utilisés de nos jours. Cette norme spécifie que :

- les noms de fichiers et de répertoires, non compris le chemin de répertoire et l'extension, ne doit pas dépasser huit caractères ;
- les noms de fichiers ne peuvent avoir qu'une seule extension et celle-ci ne doit pas dépasser trois caractères ; les noms de répertoires ne peuvent pas avoir d'extension ;

- les noms et extensions sont constitués *uniquement* des caractères A–Z, 0–9 et du caractère underscore ; les lettres minuscules sont exclues ;
- un point sépare le nom de fichier de son extension et est toujours présent même si le nom ou l’extension est vide (exemples : `FILENAME.` ou `.EXT`).
- un numéro de version, dans l’intervalle 1–32767, est ajouté à l’extension du fichier, séparé de celle-ci par un point-virgule (par exemple `FILENAME.EXT ; 1`).
- seuls huit niveaux de répertoires sont permis, en comptant le répertoire de plus haut niveau (répertoire de montage). Voir section 2.2. Ainsi, le chemin ISO-9660 le plus profond valide est :

```
texmf/L2/L3/L4/L5/L6/L7/L8/F00.BAR;1
1      2  3  4  5  6  7  8
```

Le chemin le plus profond de la TDS ne demande que sept niveaux :

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

Certains systèmes affichent un format modifié des noms ISO-9660, faisant correspondre les caractères alphabétiques avec leurs versions minuscules, supprimant le numéro de version et les points situés aux extrémités, etc. Avant la version de décembre 1996,  $\TeX$  utilisait des noms avec un mélange de casse pour les fichiers de description de fontes. Heureusement, la non distinction de cette casse n’a jamais entraîné de confusions entre fichiers. Quoi qu’il en soit, il utilise maintenant des noms avec une seule casse.

## B Problèmes d’implantation

Nous croyons que la TDS peut apporter beaucoup d’ordre à l’état vraiment anarchique de beaucoup d’installations  $\TeX$ . De plus, en indiquant un schéma commun de référence, il facilitera les tâches ennuyeuses de la documentation administrative. Enfin, il s’agit d’une partie nécessaire de tout système raisonnable permettant une véritable automatisation des distributions de packages pour  $\TeX$ .

### B.1 Adoption de la TDS

[Cette section n’est conservée que pour des raisons historiques : la TDS est maintenant assez solidement enracinée dans la plupart des distributions  $\TeX$ .] Nous reconnaissons que l’adoption de la TDS ne sera pas immédiate et universelle. De nombreux administrateurs  $\TeX$  ne seront pas enclin à effectuer le pas décisif tant que :

- des bénéfices clairs et démontrables ne seront pas visibles pour la TDS ;

- des versions compatibles TDS de tous les programmes clés ne seront pas disponibles et bien testés ;
- une période de « stabilité » n’aura pas pris place en ayant évacué les problèmes. La version publique du premier brouillon a été la première étape de ce processus.

En conséquence, la plupart des premiers essais de la TDS ont été réalisés par des membres de la commission TDS ou les développeurs de logiciels satellites de  $\TeX$ . Ceux-ci ont déjà pris place au cours de nos délibérations (voir annexe D pour un exemple d’arbre disponible par le net). Ces essais ont produits certainement un nombre substantiel de packages compatibles avec la TDS. En fait, les distributions  $\text{te}\TeX$  et  $\text{T}\TeX$  Live sont compatibles avec la TDS et sont maintenant utilisées sur de nombreux sites. Une fois que des formes prêtes à installer de packages clés compatibles avec la TDS seront plus répandues, certains administrateurs  $\TeX$  rendront leur arborescence compatible avec la TDS, éventuellement en gardant en parallèle les répertoires de production déjà existants. Ces tests permettront sans doute de faire surgir des problèmes qu’il n’était pas évident de suspecter dans le cadre confiné des sites de développement. Par exemple, cela devrait permettre de résoudre des dépendances de systèmes et de packages, d’interdépendance de packages et d’autres détails non formulés par cette version de la TDS. Après que toute la poussière sera retombée, espérons que même les plus conservateurs des administrateurs  $\TeX$  commenceront à adopter la TDS. Éventuellement, la plupart des sites  $\TeX$  auront adopté cette structure commune et la plupart des packages seront accessibles sous une forme compatible avec la TDS. Nous croyons que ce processus arrivera relativement vite. Le comité TDS couvre un large spectre d’intérêts dans la communauté  $\TeX$ . En conséquence, nous croyons que la plupart des questions clés impliquées dans la définition d’une TDS fonctionnelle ont été couvertes, souvent en détail. Les développeurs  $\TeX$  ont été consultés au sujet des problèmes d’implantation et ont essayé la structure TDS. Ainsi, nous espérons n’avoir que peu de surprises lorsque l’implantation sera mature. enfin, il y a plusieurs (actuellement ou dans le futur) éditeurs de  $\TeX$  sur CD-ROM et ceux-ci sont très motivés pour travailler sur les détails d’implantation de la TDS. Leurs productions fournissent des façons bon marché et commodes, pour les administrateurs  $\TeX$  ayant le goût de l’expérience, d’essayer la TDS.

## B.2 Compléments sur la recherche de sous-répertoires

La recherche récursive de sous-répertoires est la possibilité de spécifier une recherche non seulement sur un répertoire donné  $\langle r \rangle$  mais récursivement sur tous les répertoires situés sous  $\langle r \rangle$ . Puisque la TDS indique des emplacement précis pour la plupart des fichiers, sans aucun niveau de répertoire supplémentaire permis, une véritable recherche récursive n’est, en fait, pas exigée pour une implantation compatible TDS. Toutefois, nous recommandons fortement la recherche récursive comme



étant l'approche la plus soucieuse du confort de l'utilisateur et la plus naturelle plutôt que des méthodes tortueuses pour spécifier les chemins sans récursion. Cette propriété est déjà supportée par beaucoup d'implantations de T<sub>E</sub>X et les utilitaires compagnons comme DECUS T<sub>E</sub>X pour VMS, Dvips(k), emT<sub>E</sub>X (et ses pilotes), Public T<sub>E</sub>X, Web2C, Xdvi(k) et Y&Y T<sub>E</sub>X. La bibliothèque Kpathsea est une implantation réutilisable de la recherche de sous-répertoires pour T<sub>E</sub>X, exploitée par un grand nombre des programmes précédemment cités. Même si votre implantation T<sub>E</sub>X ne supporte pas directement la recherche de sous-répertoires, vous pouvez trouver utile d'adopter la structure si vous n'utilisez pas beaucoup de fontes et de packages. Par exemple, si vous utilisez uniquement Computer Modern et les fontes AMS, il est facile de les stocker dans une structure TDS et de lister les répertoires de façon individuelle dans des fichiers de configuration ou des variables d'environnement. Le TWG reconnaît que la recherche de sous-répertoires est une charge supplémentaire pour le système et qu'elle peut être la source d'une baisse de performance, particulièrement sur des machines lentes. Quoi qu'il en soit, nous estimons que la recherche de sous-répertoires est impérative pour une TDS bien organisée à cause des raisons évoquées à la section 2.1. Les personnes réalisant des implantations sont donc encouragées à fournir des améliorations au principe de base de la recherche de sous-répertoires afin d'éviter ces problèmes de performance, par exemple l'utilisation de cache pour les noms de fichiers (cela peut être aussi simple qu'un listing récursif de répertoires) qui sera consulté avant que la recherche sur disque ne débute. Si une correspondance est trouvée dans cette base de données, la recherche de sous-répertoires n'est plus utile et les performances seront alors indépendantes du nombre de sous-répertoires présents sur le système. Différentes implantations spécifient la recherche de sous-répertoires différemment. Pour plus de clarté, les exemples qui suivent n'utiliseront pas la fonte *(variable)*.

**Dvips** : via une variable d'environnement `TEXFONTS_SUBDIR` séparée.

**emT<sub>E</sub>X** : `t:\subdir!!`; `t:\subdir!` pour un unique niveau de recherche.

**Kpathsea** : `texmf/subdir//`

**VMS** : `texmf:[subdir...]`

**Xdvi (patchlevel 20)** : `texmf/subdir/**`; `texmf/subdir/*` pour un unique niveau de recherche. Les versions 20.50 et suivantes acceptent la notation `//`.

**Y&Y T<sub>E</sub>X** : `t:/subdir//` ou `t:\subdir\\`.

### B.3 Exemples d'arborescences spécifiques à l'implantation

La TDS ne peut pas spécifier d'emplacements précis pour les fichiers spécifiques à l'implantation tels que `texmf/ini` car un site peut avoir plusieurs implantations T<sub>E</sub>X. Cependant, pour information, nous donnons ici les emplacements par défaut pour certaines implantations. Nous apprécierons que vous nous contactiez pour

toute information supplémentaire ou correction. Ces chemins ne sont pas définitifs, peuvent ne correspondre à rien sur votre site et être modifiés sans avertissements. Nous recommandons que toutes les implantations aient des chemins de recherche par défaut qui commencent avec le répertoire courant (par exemple « . »). Permettre aux utilisateurs d'inclure le répertoire parent (par exemple « . . ») sera également utile.

### B.3.1 *AmiWeb2c 2.0*

Vous pouvez écrire à l'adresse `scherer@physik.rwth-aachen.de` pour contacter le mainteneur de cette implantation. AmiWeb2c 2 est compatible avec Web2c 7 dans toute la mesure du possible et seules les quelques différences seront décrites dans cette section. Une information détaillée sur les concepts de base est donnée dans la section sur Web2c un peu plus loin. Grâce au mécanisme SELFAUTO de Kpathsea 3.0, aucun emplacement spécifique pour l'installation d'AmiWeb2c n'est exigé tant que la structure générale de la distribution est préservée. En plus de la notation // de Kpathsea, le chemin de recherche récursive peut également débuter par  $\langle DEVICE \rangle /$ , par exemple : `TeXMF:/`, qui examinera ce périphérique spécifique de façon complète. Les binaires fournis avec la distribution AmiWeb2c sont installés dans le répertoire `bin/amiweb2c/` en dehors de l'arborescence TDS commune `share/texmf/`. En plus de cet ensemble de binaires AmiWeb2c, on trouvera deux sous-répertoires `local/` et `pastex/` avec les programmes auxiliaires. Une version allégée du système PasTeX est fournie avec AmiWeb2c (avec l'aimable autorisation de Georg Heßmann), pré-installée dans son propre répertoire `share/texmf/amiweb2c/pastex/`. Si vous voulez utiliser PasTeX, vous devez assigner le nom `TeX` : à cet emplacement. Des fichiers de documentation au format AmigaGuide devraient être stockés en `doc/guide/`, similaire à `doc/info/`.

### B.3.2 *Public DECUS T<sub>E</sub>X*

Si une autre implantation VMS, en plus de la Public DECUS T<sub>E</sub>X, existe, le nom de répertoire de premier niveau de l'implantation sera modifié en quelque chose de plus spécifique (par exemple : `vms_decus`).

```

texmf/
  vms/          fichiers spécifiques à l'implantation VMS
  exe/          commandes utilisateur finales
    common/    procédures, fichiers de définitions de commandes, etc.
    axp/       binaires exécutables pour Alpha AXP
    vax/       binaires exécutables pour VAX
  formats/     fichiers de pool, formats, bases
  help/        bibliothèque d'aide VMS et sources d'aides divers

```

---

mgr/ programmes, docs, . . . , pour la gestion du système

### B.3.3 Web2c 7

Tous les fichiers dépendants de l'implantation  $\TeX$  (.pool, .fmt, .base, .mem) sont stockés par défaut directement dans `texmf/web2c`. Le fichier de configuration `texmf.cnf` et les scripts `MakeTeX` . . . utilisés comme sous-programmes sont également stockés à cet endroit. Les fichiers spécifiques non  $\TeX$  sont stockés en suivant le codage standard de GNU. À partir d'un répertoire racine  $\langle$ préfixe $\rangle$ , nous avons les emplacements par défaut suivant :

|                              |                                                               |
|------------------------------|---------------------------------------------------------------|
| $\langle$ préfixe $\rangle/$ | racine de l'installation (/usr/local par défaut)              |
| bin/                         | exécutables                                                   |
| man/                         | pages de man                                                  |
| info/                        | fichiers info                                                 |
| lib/                         | bibliothèques (libkpathsea.*)                                 |
| share/                       | fichiers dépendants de l'architecture                         |
| texmf/                       | racine TDS                                                    |
| web2c/                       | fichiers dépendants de l'implantation (.fmt, texmf.cnf, etc.) |

Voir [http://www.gnu.org/prep/standards\\_toc.html](http://www.gnu.org/prep/standards_toc.html) pour les raisons et les descriptions de cet arrangement. Un site peut évidemment ne pas tenir compte de ces emplacements par défaut, par exemple en plaçant tout sous un seul répertoire tel que `/usr/local/texmf`.

## C Ya-t-il une meilleure solution ?

La définition de la TDS a nécessité de nombreux compromis. La structure générale ainsi que les détails relatifs à chaque répertoire ont été arrêtés en trouvant un terrain d'entente entre de nombreuses opinions. Les idées directrices étaient la faisabilité (en termes de ce qui pouvait techniquement être réalisé et ce que l'on pouvait raisonnablement attendre des développeurs) et la régularité (les fichiers devant être regroupés selon une disposition qui « avait un sens »). Certaines idées intéressantes n'ont pas pu être appliquées à cause du manque de support nécessaire des implantations :

- Contrôle du chemin de recherche au niveau de  $\TeX$ . Si les documents pouvaient restreindre la recherche de sous-répertoires à un certain sous-répertoire par l'intermédiaire d'une certaine syntaxe portable au niveau des noms de fichiers, les restrictions sur l'unicité des noms de fichiers auraient pu être considérablement relâchées (avec la participation des formats) et les chemins de recherche de  $\TeX$  n'auraient pas eut besoin de dépendre du format.

- Plusieurs arborescences logiques `texmf`. Par exemple, un site aurait pu avoir un emplacement (en lecture seule) pour les fichiers stables et différents emplacements (accessibles en écriture) pour les fontes créées de façon dynamique et d'autres fichiers. Il aurait été raisonnable pour deux tels arbres d'être logiquement assemblés lors des recherches. On se référera à l'article de Michael Downes dans les références pour voir comme cela peut fonctionner en pratique avec Web2C.

## C.1 Structure pour les macros

Le TWG s'est mis d'accord sur la disposition  $\langle format \rangle / \langle package \rangle$  après de longues discussions sur la meilleure façon de placer les fichiers. L'autre choix principal à cette disposition était le schéma qui inversait l'ordre de ces répertoires :  $\langle package \rangle / \langle format \rangle$ . Ce renversement avait un grand intérêt : il laissait tous les fichiers relatifs à un package particulier au niveau d'un emplacement unique. L'arrangement effectivement adopté tend à disperser les fichiers au niveau de deux ou trois emplacements (macros, documentations et fontes, par exemple, sont dispersés dans différentes directions de l'arbre dès le premier niveau). Cependant, la structure  $\langle format \rangle / \langle package \rangle$  a gagné pour deux raisons :

- C'est plus proche de la pratique courante. En réalité, plusieurs membres du TWG avait déjà implanté la hiérarchie TDS. La seconde solution n'avait été utilisée sur aucun des sites connus et le TWG avait le sentiment qu'imposer quelque chose qui n'avait aucune expérience pratique serait une erreur.
- La seconde façon de faire augmente le nombre de répertoires de premier niveau et ainsi, les fichiers aurait dû être trouvés avec une recherche de sous-répertoires devant effectuer sa propagation dans un arbre large et peu profond. Cela aurait pu avoir un profond impact au niveau de l'efficacité de la recherche de sous-répertoires.

## C.2 Structure des fontes

Le TWG s'est plus disputé à propos des fontes que pour n'importe quel autre sujet. Ce n'est pas surprenant : la nécessité d'utiliser les fontes PostScript, en pleine prolifération, avec  $\TeX$  a rendu l'arrangement précédent avec tous ses fichiers dans un seul répertoire intenable et, par conséquent, a initié l'effort de la TDS.

### C.2.1 Emplacement par type de fichier de fontes

Nous avons considéré l'arrangement avec le fournisseur en premier, arrangement utilisé sur de nombreux sites :

```
texmf/fonts/⟨fournisseur⟩/⟨famille⟩/⟨type⟩/
```

Cela facilite la maintenance de l'arborescence des fontes puisque tous les fichiers appartenant à une famille donnée étaient au même emplacement. Mais, à moins que tous les programmes effectuant des recherches dans cet arbre n'emploient une certaine forme de cache, il risque d'y avoir de sérieux problèmes de performance. Par exemple, afin de trouver un fichier TFM, la plus simple des implantations demandera à T<sub>E</sub>X de rechercher à travers tous les sous-répertoires qui contiennent les fichiers PK dans tous les modes et à toutes les résolutions. Finalement, un sondage auprès de développeurs a révélé une résistance considérable à l'implantation de suffisamment de mécanismes de cache. Ce schéma a donc été abandonné. Le schéma de la TDS permet au moins à la recherche arborescente de se limiter à un type correct de fichier. Des réserves à propos de l'efficacité demeurent mais il semble qu'on ne puisse pas mieux faire sans abandonner entièrement la recherche récursive. Nous avons également pensé à regrouper tous les fichiers de fontes par type de fichier afin que les sources METAFONT soient dans un répertoire `texmf/fonts/mf`, les fichiers de liste de propriétés dans `texmf/fonts/pl`, les formes diverses des fontes Type 1 séparées, et ainsi de suite. Bien qu'évidemment plus consistante, nous avons pensé que le revers de la médaille consistant en une plus grande complication au niveau de la constructions des chemins était plus important. La TDS rassemble quelques types de fichiers (`mf` et `pl` sous `source`, `pfa`, `pfb` et `gsf` sous `type1`) là où nous pensons que c'est avantageux.

### C.2.2 Emplacement du mode et de la résolution

Nous avons considéré placer le mode au plus profond de l'arborescence des fontes :

```
texmf/fonts/pk/<fournisseur>/<famille>/<mode>/<dpi>/
```

Dans ce cas, cependant, il est difficile de limiter la recherche de sous-répertoires au mode nécessaire à un périphérique particulier. Nous avons alors pensé à déplacer le `dpi<nnn>` au dessous du mode :

```
texmf/fonts/pk/<mode>/<dpi>/<fournisseur>/<famille>/
```

Mais alors il n'est plus possible d'omettre le niveau `dpi<nnn>` pour les systèmes qui peuvent choisir d'utiliser les noms de fichiers longs.

### C.2.3 Fontes bitmap sans mode

La TDS demande d'utiliser un répertoire unique `modeless/` comme le nom de mode pour les utilitaires qui produisent des fontes bitmap, par exemple `texmf/fonts/modeless/times/`. Cela a le considérable avantage de ne pas exiger que chaque nom de répertoire de ce type soit listé dans le chemin de recherche. Une autre possibilité était d'utiliser le nom de l'utilitaire au-dessous duquel tous les répertoires

auraient été rassemblés. Cela a l'avantage de séparer, disons, les fontes bitmap produites par `gsftopk` de ceux produits par `ps2pk`. Cependant, nous avons estimé que ce n'était pas nécessaire, la plupart des sites n'utilisant qu'un seul programme dans ce but. Ainsi, les fontes PK et GF identifient généralement leur créateur dans les commentaires de fonte suivant l'octet PK\_ID. Nous supposons implicitement que METAFONT était le seul programme produisant des fontes bitmap dépendant du mode. Si cela devait se révéler faux, nous pourrions ajouter une abréviation pour le programme au nom de mode, comme dans `mfcx` et `xyzcx` pour un hypothétique programme `Xyz`, ou bien nous pourrions à ce moment ajouter un niveau de nom de programme supplémentaire de façon uniforme sur l'arbre. Il semblait plus important de représenter de façon concise la situation actuelle que de s'embêter à propos d'hypothétiques possibilités qui pourront ne jamais exister.

### C.3 Structure de documentation

Nous avons pensé à placer les fichiers de documentation supplémentaires dans le même répertoire que les fichiers sources pour les packages mais nous pensons que les utilisateurs seront capables de trouver la documentation séparément des sources puisque la plupart de ceux-ci ne s'intéressent pas aux sources. Nous espérons qu'une structure séparée, mais parallèle, pour la documentation devrait : (1) laisser toute la documentation ensemble et (2) rendre aussi directe que possible la recherche d'une documentation particulière par l'utilisateur.

## D Références

Cette annexe donne des pointeurs sur des fichiers et autres documents ayant rapport avec la TDS. Pour les références `abbrctan`, nous utilisons `http://www.ctan.org` comme domaine uniquement pour être certain que les liens seront toujours valides dans ce document. Voir `http://www.ctan.org/tex-archive/CTAN.sites` pour une liste complète des sites CTAN : il y a des miroirs partout dans le monde.

- Ce document, dans plusieurs formats (tex, dvi, info, pdf) :  
`http://tug.org/tds/`
- Les archives de la liste de diffusion TDS :  
`http://tug.org/mail-archives/twg-tds/`
- Le pilote DVI niveau 0 standard :  
`http://www.ctan.org/tex-archive/dviware/driv-standard/level-0/`
- *Filenames for  $T_{E}X$  fonts*, avec les listes des noms recommandés pour les fournisseurs et les familles :  
`http://tug.org/fontname/`
- Le système de fichiers standard sur CD-ROM ISO-9660 :  
`http://www.iso.ch/cate/cat.html`

- *Components of  $\TeX$* , un papier de Joachim Schrod :  
<http://www.ctan.org/tex-archive/documentation/components-of-TeX/>
- *Managing Multiple TDS trees*, un article de Michael Downes :  
<http://tug.org/TUGboat/Articles/tb22-3/tb72downes.pdf>
- Un ensemble complet de modes METAFONT :  
<http://www.ctan.org/tex-archive/fonts/modes/modes.mf>
- Une grande collection de bases de données et de styles BIB $\TeX$  :  
<ftp://ftp.math.utah.edu/pub/tex/bib/>

## E Collaborateurs

Le TWG n'a eu aucune réunion physique : le courrier électronique était le support de communication. Sebastian Rahtz est le responsable du conseil technique du  $\TeX$  Users Group. Norman Walsh en est le président fondateur. Karl Berry est le rédacteur actuel. La liste des collaborateurs est devenue trop importante pour pouvoir honnêtement l'inclure, certains d'entre eux auraient inmanquablement été oubliés par inadvertance. Vous pouvez considérer les archives des listes de diffusion [tds@tug.org](mailto:tds@tug.org) et [tex-live@tug.org](mailto:tex-live@tug.org) comme étant l'enregistrement des collaborations.

## Postface de Fabrice Popineau

L'article qui précède est la traduction fidèle d'une norme qui a ses origines dans les années 1995 et qui est pratiquement figée depuis 1999. Une norme n'est que factuelle et on est en droit de se demander quelles motivations ont bien pu conduire à un arrangement des fichiers aussi compliqué dans l'arborescence des fichiers  $\TeX$ .

J'ai été amené depuis environ huit ans à côtoyer la TDS de près, puisque je me suis occupé de porter le code source de  $\TeX$  utilisé à la fois par  $\text{te}\TeX$  et par  $\text{TeXLive}$  sous Windows. Je pense donc pouvoir apporter un éclairage certainement subjectif mais pragmatique sur les avantages et les inconvénients de cette norme.

Quelles étaient les distributions  $\TeX$  disponibles à l'époque où la TDS est née ? Il existait des distributions pour les grands systèmes (Unix, VMS ou autres), en Pascal et parfois en C et des distributions pour ordinateurs personnels, la plus célèbre étant probablement  $\text{em}\TeX$ . Il est à noter que  $\text{em}\TeX$  n'offrait pas de recherche récursive dans les répertoires. Il n'y avait pas de distributions unifiées comme ce peut être le cas aujourd'hui où il ne reste que deux distributions libres majeures.

La multiplication des contributions à T<sub>E</sub>X et L<sup>A</sup>T<sub>E</sub>X a fait naître un besoin : celui d'un schéma de rangement pour les trop nombreux fichiers. Ce schéma de rangement devait présenter deux qualités : premièrement, avoir un sens humainement parlant et deuxièmement *permettre à la machine de retrouver les fichiers rapidement*.

Ceci dit, l'existant consistait en une multitude de systèmes d'exploitation avec des caractéristiques assez différentes en termes de systèmes de fichiers, aussi bien du point de vue des possibilités que de celui des performances.

Les premiers graveurs de CD-ROM faisant leur apparition, l'éventualité d'une distribution T<sub>E</sub>X sur CD-ROM était à envisager et le média était nouveau par ses caractéristiques. Le CD-ROM fournissait un nouveau type de système de fichiers qui avait le mauvais goût de ne même pas offrir les mêmes fonctionnalités sous tous les systèmes d'exploitation. Les extensions RockRidge et Joliet sont venues y palier par la suite.

Si je rappelle ces éléments d'histoire, c'est parce qu'il est primordial de comprendre que les décisions qui ont présidé à la définition de la TDS ont été basées sur des contraintes techniques qui datent de cette époque (circa 1994).

En particulier, deux choix importants ont été faits sur la base suivante :

- Les CD-ROM ne permettaient de stocker de manière universelle que des noms au format 8.3, donc il fallait que la TDS respecte cette limitation. On a donc introduit la dénomination `600dpi/foo.pk` pour le fichier bitmap de la police `foo` à la résolution de 600dpi. Une limitation similaire concernait la profondeur des répertoires qui ne devait pas dépasser huit niveaux à partir de la racine.
- Les systèmes de fichiers dont on disposait à l'époque (VAX et Unix) offraient de piètres performances dès qu'il y avait trop d'entrées dans un répertoire donné. On a cherché une solution qui limitait ce nombre d'entrées par répertoire, et on a donc décidé de ne pas stocker tous les fichiers d'une police donnée dans un même répertoire mais de les répartir dans différents répertoires selon leur type (`mf`, `tfm`, `pk`, etc.).

Il est bien clair que cinq ou six ans après, ces limitations ont volé en éclats.

Quels sont les bénéfices de la TDS aujourd'hui ?

D'un point de vue strict, je dirais très faibles. La TDS laisse comme non-spécifiés des points critiques : que faire si un nom de fichier est dupliqué dans l'arborescence ? Il y a même des fichiers pour lesquels la TDS ne dit pas explicitement où les mettre, comme par exemple les fichiers de documentation de polices dont on ne sait pas si ils doivent être dans `texmf/doc/fonts` ou dans `texmf/doc/fonts/<supplier>`. La TDS fournit un ensemble de règles qui permet à tout un chacun de ranger ses fichiers d'une façon qui ressemblera à celle d'un collègue qui respecterait également la TDS, mais elle ne prétend pas contrôler complètement le système.

Les bénéfices nets sont donc faibles pour les raisons suivantes :



- l'agencement des fichiers est complexe,
- la mise à la forme TDS d'un package requiert une quantité de manipulations,
- les distributions n'offrent pas d'outil pour manipuler un package dans son ensemble... et pour cause : la TDS ne définit pas strictement la notion de package.

Du point de vue de l'utilisateur, le seul bénéfice réside dans le fait qu'une norme contrôlant en partie le système est peut-être préférable à une anarchie totale. Mais la situation est loin d'être idéale.

Au vu de ce constat, quelles solutions s'offrent à nous? On pourrait souhaiter tout remettre à plat pour obtenir un rangement plus rationnel (humainement parlant) des fichiers. Par exemple mettre les différents fichiers de la police times dans `texmf/fonts/times/{afm, tfm, type1, vf}/`. Il est clair que la manipulation *manuelle* des packages en serait simplifiée et la lisibilité de la structure également. On peut procéder de la même façon pour les packages  $\LaTeX$ . Seul le fichier `texmf.cnf` est à modifier. Il est ainsi possible de construire une répartition des fichiers alternative à la TDS qui suffirait certainement à prendre en compte  $\LaTeX$  et  $\ConTeXt$ . En revanche, un niveau de complication supplémentaire apparaîtra forcément dès que l'on voudra autoriser qu'un format comme  $\LaTeX$  soit généré au-dessus de  $\pdf\TeX$  ou au-dessus de Oméga, prenne en compte des options telles que  $\enc\TeX$  ou non.

Dans le cadre du  $\TeX$ Live, il serait vain d'espérer convaincre l'ensemble des contributeurs de lâcher la TDS. Pourtant, la situation n'est pas désespérée. Il est possible de transformer la TDS en un outil à forte valeur ajoutée. Voici comment.

Tout part de la remarque concernant les outils pour manipuler les packages dans leur ensemble. Quel utilisateur a besoin de savoir que le package `tabvar` contient quinze fichiers qui seront dispersés dans huit répertoires différents? Personne. En fait, l'objectif à atteindre, c'est masquer cette complexité à l'utilisateur. À partir du moment où l'on dispose d'outils pour manipuler globalement les packages, tous les jeux de règles sont équivalents.

Que faudrait-il faire pour disposer de ces outils? Réponse : modifier un tout petit peu la TDS pour la rendre plus stricte sur certains points. En ajoutant quelques règles à la norme, on peut arriver à une situation où la gestion automatique des packages devient possible : ajout, mise à jour, suppression, validation. En quelques mots, voici comment combler les lacunes :

1. unicité des noms de packages : il faut garantir que chaque nom de package reste unique ; un nom attribué à un auteur devient indisponible pour les autres.
2. il faut garantir que deux packages n'utilisent pas les mêmes répertoires et que tout répertoire ne contient que des répertoires ou que des fichiers.
3. pour respecter la règle précédente, il faut associer des noms prédéfinis aux répertoires qui contiendront les fichiers des packages associés à des formats  $\TeX$ .

Par exemple, les fichiers du package `latex` seront trouvés dans les répertoires `texmf/tex/latex/base` et `texmf/tex/latex/config`.

4. en définissant des listes de

- moteurs  $\text{T}_{\text{E}}\text{X}$  (`tex`, `etex`, `omega`, ...),
- formats  $\text{T}_{\text{E}}\text{X}$  (`plain`, `latex`, `context`, ...),
- vendeurs de polices,
- types de fichiers de police (`pk`, `type1`, `tfm`, `vf`, ...),
- langages (pour les fichiers de documentation),

il est possible de définir un ensemble fini de règles permettant de déterminer un ensemble de répertoires à partir d'un nom de package. Cet ensemble de répertoires définit les endroits où les fichiers de ce package peuvent et doivent se trouver. C'est l'espace privé du package. Aucun fichier d'un autre package ne pourra s'y trouver.

J'ai testé ce schéma sur la prochaine version du  $\text{T}_{\text{E}}\text{X}$ Live. Plus de 80 % des packages, soit tous ceux qui ne sont pas dépendants de la distribution Web2C, sont gérés automatiquement par ce système. La bibliothèque Perl de gestion des fichiers `.tpm` de méta-données fait 2500 lignes. Avec cet outil, on peut reconstruire ou valider une arborescence `texmf` qui respecte une version plus stricte de la TDS. Maintenant, on pourrait songer sérieusement à équiper CTAN d'un tel outil : CTAN pourrait valider la structure des packages soumis, intégrer les packages valides à sa base de données, et les délivrer sur demande sous une forme prête à installer. Les utilisateurs seraient enfin délivrés du besoin de comprendre la TDS et la TDS(+) servirait à nouveau à quelque chose.

Fabrice Popineau