

# *Cahiers* **GUT**enberg

☞ COMMENTAIRES SUR LA PORTABILITÉ  
DE T<sub>E</sub>X

☞ Daniel TAUPIN

*Cahiers GUTenberg*, n° 15 (1993), p. 3-31.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_1993\\_\\_15\\_3\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_1993__15_3_0)>

© Association GUTenberg, 1993, 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.



# Commentaires sur la portabilité de T<sub>E</sub>X

---

Daniel TAUPIN

*Laboratoire de Physique des Solides  
associé au CNRS  
Bâtiment 510, Centre Universitaire  
F - 91405 ORSAY Cedex  
taupin@frups51.bitnet  
taupin@rsovax.ups.circe.fr*

## Avertissement

Cet article est publié ici sous la seule responsabilité de son auteur et ne saurait représenter en quoique ce soit l'opinion officielle de l'association GUTenberg ni de la rédaction des *Cahiers GUTenberg*. Ces *Cahiers* publieront d'ailleurs des réactions à cet article en les regroupant dans un prochain numéro.{Ndlr}

*This paper is published under the own responsibility of the author. It does not represent the official opinion neither of the GUTenberg association nor of the editorial board of Cahiers GUTenberg. These Cahiers intend to publish further reactions and contradictory papers in a forthcoming issue.*{From the editorial board}

## Résumé

T<sub>E</sub>X est un logiciel réputé, au moins dans le monde universitaire, pour sa qualité d'impression, sa versatilité pour l'usage de langues variées – y compris dans un même texte – et pour sa portabilité. Malheureusement, le développement récent de « styles » nationaux, de polices avec lettres accentuées, et la saisie à partir de claviers autres que le jadis tout-puissant QWERTY réduisent considérablement cette portabilité si l'on n'y prend garde. Nous proposons donc aux auteurs de fichiers source T<sub>E</sub>X un certain nombre de règles – voire d'« interdits » – assurant la portabilité entre modèles d'ordinateurs et formats de base distincts :

- n'expédier que des caractères ISO 8859 (ASCII) à 7 bits et imprimables,
- expédier tous les fichiers de styles et de macros hormis standards internationaux PLAIN et L<sup>A</sup>T<sub>E</sub>X,
- n'expédier que des textes compatibles sans modification avec les formats basés sur n'importe quelle série de polices, tout particulièrement ceux basés sur les polices **cmxxx**,

- renoncer définitivement à expédier des textes qui utilisent les coupures automatiques de mots, hormis l’anglais.

Nous formulons aussi un certain nombre de critiques et de recommandations aux auteurs de styles et implémenteurs de formats de base PLAIN et L<sup>A</sup>T<sub>E</sub>X.

### Abstract

T<sub>E</sub>X is a world known software, and usually favourite at least in the university or research world. This favour is due to its printing quality, its adaptability to various languages – including in the same paper – and to its portability. Unfortunately, the recent development of national “styles”, of fonts with accented letters, and its typing from keyboards differing from the previously allmighty QWERTY ones actually restrict that portability, unless extreme care is exercised. Therefore, we propose the T<sub>E</sub>X file writers a certain number of rules – even some “interdictions” – to ensure a satisfactory portability between the various base formats and computer types :

- send only 7 bit ISO 8859 (ASCII) files, containing only printable characters,
- send all macro and style files except world international standards PLAIN and L<sup>A</sup>T<sub>E</sub>X,
- only send texts which can be typeset with formats based on any of the various standard font series, especially those based on **cmxxx** fonts,
- do not rely on automatic hyphenation of non english texts.

We also express some criticism and further recommendations to style writers and base format (PLAIN and L<sup>A</sup>T<sub>E</sub>X) implementors.

## 1. Le plus grand mérite de T<sub>E</sub>X...

Il est évidemment inutile de vanter les mérites de T<sub>E</sub>X dans une revue dédiée, comme celle-ci, aux usagers, et même aux « amateurs » voire aux « amoureux » de ce logiciel. Toutefois, au moins à titre de rappel nous mentionnerons :

1. C’est un logiciel gratuit, aux frais de duplication et d’envoi près.
2. Il produit une impression dont la qualité laisse assez loin derrière la plupart des autres logiciels de PAO, à l’exception peut-être de ceux qui pilotent les *photocomposeuses* des imprimeries professionnelles de haut de gamme.

3. Grâce à son logiciel associé METAFONT il peut s'enrichir à des coûts de main d'œuvre modérés de nouvelles polices, notamment de *pictogrammes*.

En fait il suffit de discuter avec des informaticiens du secteur industriel ou des publieurs de rapports extérieurs au monde T<sub>E</sub>X pour se rendre compte que les arguments précédents ne sont recevables que... pour ceux qui sont *déjà* convaincus de la supériorité de T<sub>E</sub>X. Et les raisons – en grande partie irrationnelles – sont les suivantes :

- Dans le monde des entreprises visant à faire du *profit*<sup>1</sup> les logiciels *gratuits* sont *suspects*, parfois d'être infectés de *virus*, en tous cas de manquer... aux règles élémentaires des affaires.
- Hormis le monde universitaire – au sens large du terme – traditionnellement réticent vis-à-vis du monde du *business*, vivant à l'économie financière<sup>2</sup> et où certaines revues ont imposé T<sub>E</sub>X pour la soumission des manuscrits<sup>3</sup>, le monde des secrétariats et des publieurs de rapports – voire de *thèses* – ne connaît que les logiciels de PAO :
  - dont font usage les collègues de même catégorie professionnelle,
  - dont ils (ou leurs supérieurs) reçoivent la publicité, les annonces de stages, des démarchages, etc.
- Enfin, *last but not least*, l'opinion que la plupart des usagers ont de leurs logiciels est que « le meilleur logiciel est celui que j'utilise ». On peut ajouter : « par définition », en sous-entendant perfidement que personne n'avoue jamais *avoir fait le mauvais choix*, fût-ce collectivement, et en remarquant avec plus de sérénité que le temps *investi* à acquérir l'expérience d'un logiciel justifie qu'on n'en change qu'en cas d'absolue nécessité.

Comme d'autre part les évaluations que l'on fait de la qualité ou de la beauté – typographique, littéraire, musicale, picturale, architecturale ou paysagique – est avant tout culturelle, les adeptes des PAO les plus

---

1. Terme employé ici sans aucune connotation de valeur.

2. Souvent au prix de grosses dépenses en main d'œuvre non comptabilisée dans les coûts.

3. *Publish or perish...*

pauvres<sup>4</sup> font « de nécessité vertu » : ainsi les polices associées à Word ou Wordperfect deviennent le parangon de la typographie, la majorité des non-professionnels de la composition de texte proclame qu'on *ne doit pas accentuer les majuscules* et utilise le soulignage là où tous les typographes changeraient de police. *Et cætera*.

## 2. ... c'est sa portabilité

Il y a en revanche un domaine où la supériorité de T<sub>E</sub>X est rarement contestée et difficilement contestable, au moins en l'état actuel des autres logiciels, à savoir sa *portabilité*<sup>5</sup>, qui tient aux raisons suivantes :

1. T<sub>E</sub>X n'est pas lié à un constructeur ou à producteur de logiciel, de sorte qu'il tourne sur presque tous les ordinateurs utilisables pour la PAO, depuis IBM (sous VM ou MVS) jusqu'aux systèmes UNIX en passant par les PC et les Macintosh.
2. Les fichiers « source » peuvent ne se composer que de caractères ASCII visibles (donc éditables) et peuvent par conséquent être transportés d'un ordinateur à n'importe quel autre sans grand dommage, et ils peuvent être « édités<sup>6</sup> » avec n'importe quel *éditeur* tel que EMACS, les éditeurs incorporés à *Turbo-C* et *Turbo-Pascal*, EDT ou n'importe quel « EDIT ». Pour les mêmes raisons, n'importe quel programme écrit en C, en Pascal, en Fortran ou en ADA peut facilement produire un fichier de résultats compilable par T<sub>E</sub>X pour une impression de qualité.

Après avoir ainsi chanté la gloire de la portabilité de T<sub>E</sub>X, on est cependant obligé de constater que la réalité n'est pas tout-à-fait aussi reluisante que les pétitions de principe laudatives, à cause :

- des diversités de versions du moteur<sup>7</sup> T<sub>E</sub>X, à savoir V2.x<sup>8</sup>, V3.x et MLT<sub>E</sub>X,

---

4. « Pauvres » s'applique à PAO, mais si vous voulez appliquer l'épithète à *adeptes*, vous en avez le droit.

5. Encore que je me sois fait rétorquer que le respect des droits des auteurs (*copyright* en français) rendaient la portabilité des textes plus condamnable qu'utile!

6. Au sens anglo-ordinal du terme.

7. Pour ne pas dire « le compilateur »...

8. En principe obsolète, mais nos enquêtes montrent encore 10 % de T<sub>E</sub>X version 2.x encore en activité.

- des diversités de polices disponibles,
- des variations de formats de base PLAIN et L<sup>A</sup>T<sub>E</sub>X,
- des variétés de codages des caractères  $\geq 128$  (caractères ISO 8859, codage 850 des PC, codage spécifiques Macintosh),
- de l'incroyable diversité des logiciels graphiques, dont aucun n'est accepté par *tous* les pilotes (*drivers*) d'imprimante. Certes ceci n'est pas strictement lié à T<sub>E</sub>X, mais l'incorporation de *figures* est un problème incontournable et la portabilité d'un T<sub>E</sub>Xte<sup>9</sup> implique aussi la portabilité des figures.

### 3. La portabilité, à quel niveau ?

On l'oublie parfois, mais la portabilité de T<sub>E</sub>X se situe à deux niveaux :

1. la portabilité du fichier source (le plus souvent appelé \*.TEX),
2. la portabilité du fichier résultat produit par le moteur T<sub>E</sub>X, le DVI<sup>10</sup>, lequel est enfin visualisé ou imprimé par un pilote qui, pour sa part, dépend crucialement de l'ordinateur sur lequel il tourne et du périphérique destinataire.

Si la portabilité du fichier source est une nécessité communément admise, celle du DVI n'est pas sans importance. En effet beaucoup de logiciels associés à T<sub>E</sub>X sont livrés avec leur documentation sous forme d'un DVI qu'il faut pouvoir imprimer ou au moins visualiser localement<sup>11</sup>. De même, à défaut d'être certains de disposer exactement des mêmes versions de formats PLAIN ou des mêmes versions de styles L<sup>A</sup>T<sub>E</sub>X, des imprimeurs peuvent préférer qu'on leur communique un DVI et les polices plutôt que de tomber sur des aléas de compilation ou de classement d'index.

---

9. Nous nous sommes permis de forger ce néologisme pour dire brièvement : « un texte codé en langage T<sub>E</sub>X », par analogie aux *T<sub>E</sub>Xperts* inventés par D. KNUTH[1]. De même, nous nous permettons d'utiliser le vocable *L<sup>A</sup>T<sub>E</sub>Xte* pour : « un texte codé en L<sup>A</sup>T<sub>E</sub>X ».

10. Ce qui veut dire « *device independent* », ne l'oublions pas.

11. Cas de divers logiciels diffusés depuis l'Allemagne, notamment ceux d'Eberhardt MATTES ou de Friedrich SOWA (**bm2font**).

## 4. Les défauts de portabilité de T<sub>E</sub>X

### 4.1. Les défauts possibles de portabilité du T<sub>E</sub>Xte source

Sachant que la perfection n'est toujours pas de ce monde, il convient de dire que rien n'est jamais rigoureusement portable, sauf à utiliser exactement le même ordinateur, la même version du moteur T<sub>E</sub>X, le même pilote et la même imprimante.

Mais certains défauts de portabilité sont mineurs, comme par exemple le positionnement exact des accents selon qu'on utilise des polices réelles ou virtuelles<sup>12</sup>, ou bien de rares coupures de mots malencontreuses ; d'autres sont catastrophiques, par exemple quand le moteur T<sub>E</sub>X se bloque en bouclant sur un diagnostic ou quand des aléas de police rendent franchement *fausse*<sup>13</sup> une formule mathématique... sans donner le moindre diagnostic.

Bien plus grave est le cas de deux systèmes T<sub>E</sub>X tournant sur des ordinateurs différents – et dont il n'est par conséquent pas question d'exiger le remplacement – ont des comportements irréductiblement différents. C'est par exemple le cas si l'un des moteurs est OzT<sub>E</sub>X 1.4.1 qui ne possède ni le traitement des polices virtuelles<sup>14</sup> ni la conversion de *code page*, ce qui de proche en proche conduit à des comportements « déviants » quoique traités au mieux dans Euro-OzT<sub>E</sub>X pour avoir des coupures convenables des mots accentués.

Plus grave encore, nous connaissons deux implémentations de L<sup>A</sup>T<sub>E</sub>X – l'une étant Euro-OzT<sub>E</sub>X, l'autre un MLT<sub>E</sub>X tournant sur SUN – qui imposent l'usage de NFSS<sup>15</sup>. NFSS est un logiciel performant, mais son défaut est d'avoir modifié *sans prévenir* l'effet des commandes de changement police : `\bf`, `\tt`, `\sl`, etc. Si NFSS était un « style » ou une « option de style » ce serait sans gravité, mais l'ennui est que NFSS est usuellement introduit dans un format standard, s'appelant L<sup>A</sup>T<sub>E</sub>X sans signe distinctif. De ce fait, quand on transmet un L<sup>A</sup>T<sub>E</sub>Xte à un correspondant inconnu, on ne peut savoir à l'avance – peut être lui-même ne le sait-il pas – si ce texte sera compilé dans l'environnement NFSS ou en L<sup>A</sup>T<sub>E</sub>X standard. L'accumulation de ces trois erreurs – la plus grave étant l'abus du sigle L<sup>A</sup>T<sub>E</sub>X sans signe distinctif

---

12. Les accents obtenus d'après les polices virtuelles `dmxxx` sont plus hauts que ceux élégamment cadrés des polices `dcxxx`.

13. Voir exemple plus loin.

14. OzT<sub>E</sub>X est un ensemble [moteur T<sub>E</sub>X+pilote].

15. *New Font Selection Scheme*, écrit par Frank MITTELBACH et Rainer SCHÖPF.

– conduit à un comportement erratique au sens de la portabilité... et dont l’auteur de L<sup>A</sup>T<sub>E</sub>Xtes portables devra soigneusement se prémunir.

Enfin se pose le problème des jeux de polices disponibles et, surtout, celles « programmées » par défaut dans les formats PLAIN et L<sup>A</sup>T<sub>E</sub>X. Alors, pour peu que l’auteur du T<sub>E</sub>Xte ait besoin de spécifier explicitement un choix de police (`cmxxx`, `dmxxx`, `dcxxx`, bientôt `ecxxx`)<sup>16</sup>, son ouvrage verra sa portabilité s’effondrer lamentablement.

## 4.2. Les défauts possibles de portabilité du DVI

Contrairement à ce qu’on pourrait penser *a priori*, les DVI sont généralement bien plus portables que les fichiers source T<sub>E</sub>X. La seule condition est que le destinataire possède les mêmes polices<sup>17</sup> ou qu’il soit capable de les fabriquer par METAFONT. En conséquence, si le problème de compatibilité des polices reste entier<sup>18</sup> – et même si le document initial n’y fait pas de référence explicite – le problème de la compatibilité des moteurs T<sub>E</sub>X, des formats PLAIN, L<sup>A</sup>T<sub>E</sub>X ou des styles est évacué.

## 5. Les remèdes pour une meilleure portabilité de T<sub>E</sub>X

Pour permettre le portage correct d’un T<sub>E</sub>Xte entre plusieurs implémentations, on peut *a priori* s’orienter vers deux « politiques » extrêmes et fondamentalement opposées :

1. contraindre toutes les installations T<sub>E</sub>X (et pilotes annexes) à être compatibles,
2. contraindre l’auteur du T<sub>E</sub>Xte à n’écrire que des choses portables *partout*.

En fait, aucune de ces deux solutions n’est – on le verra – strictement praticable. Ceci nous amènera, outre des recommandations aux auteurs

---

16... ou qu’il veuille des majuscules grecques grasses dans ses formules en les préfixant « bêtement » par `\bf` – voir plus loin 6.4.2.

17. Même nom, même grandissement et même *check sum* du TFM.

18. Les fichiers de polices T<sub>E</sub>X sont cependant presque toujours portables... à condition de les « porter » en *bin aire*.

« ordinaires » de T<sub>E</sub>Xtes, à émettre quelques requêtes aux inventeurs de styles ou de variantes de formats de base, notamment en ce qui concerne les polices et les coupures automatiques de mots.

### 5.1. Rendre les implémentations de T<sub>E</sub>X compatibles?

La première politique, rendre compatibles toutes les installations de T<sub>E</sub>X, est de toute évidence le souci de Donald K<sub>N</sub>UTH quand il s'oppose à ce qu'on change quoi que ce soit à la définition de T<sub>E</sub>X. On lui obéit par respect des règles de droit moral<sup>19</sup>, mais la raison profonde est le maintien de la portabilité entre les T<sub>E</sub>X.

Bien sûr, si un T<sub>E</sub>Xte parfaitement *standard*<sup>20</sup> donne des résultats gravement erronés, on est en droit de considérer qu'il y a une « bogue » dans le moteur ou le format de base de l'installation réceptrice et on ne peut faire mieux que d'espérer l'installation d'une nouvelle version plus fiable ou moins obsolète. Mais tout dépend alors des capacités du destinataire à mettre à jour ou à obtenir des gourous locaux cette mise à jour du moteur T<sub>E</sub>X ou des formats de base. Autant dire que s'il s'agit d'une « bogue » fréquente il vaut mieux se référer à la loi de MURPHY – *that's not a bug, that's a feature* – et chercher un contournement qui l'évite.

On peut aussi espérer que les dix pour cent<sup>21</sup> des moteurs T<sub>E</sub>X restant encore en version V2.x passent enfin à la version V3.x – sachant que K<sub>N</sub>UTH a autorisé à titre exceptionnel cette mise à jour de T<sub>E</sub>X – mais il ne faut pas se faire d'illusion : la moitié du monde informatisé, c'est-à-dire *grosso modo* la rive ouest de l'Atlantique sauf le Québec, vit très bien avec le code ASCII à 26 lettres non accentuées et on ne voit pas quel intérêt immédiat ce demi-monde aurait à se soucier d'installer le jeu des polices **dcxxx** (une demie douzaine de mégaoctets<sup>22</sup> sans compter les tracas de formats PLAIN ou L<sup>A</sup>T<sub>E</sub>X) ou même de faire installer les pilotes acceptant les « polices virtuelles ». D'ailleurs les réponses au questionnaire test précédemment cité sont instructives : au moins 90 % des correspondants ont les polices **cmxxx**

---

19. Le *droit moral* est – dans la plupart des législations occidentales – le droit qu'a l'auteur d'une œuvre ou ses ayant-droit de ne pas voir celle-ci *dénaturée*. Ce droit est indépendant du fait que l'œuvre est diffusée gratuitement ou moyennant finances. En théorie, ce droit est éternel.

20. C'est-à-dire codé conformément aux spécifications du T<sub>E</sub>Xbook[1].

21. Selon un questionnaire envoyé par courrier électronique à quatre listes d'utilisateurs T<sub>E</sub>X, et qui a fourni plus de 60 réponses sous forme de test.

22. En moyenne et en résolution 300 bits par inch.

comme polices standard en PLAIN T<sub>E</sub>X (pas MLT<sub>E</sub>X) ce qui exclut qu'ils puissent couper automatiquement les mots à lettres accentuées.

Inévitablement, quand on parle de *mise à jour* on pense « norme » (sous-entendu au singulier); il faut cependant prendre conscience que les « normes » ont perdu leur caractère obligatoire pour n'être plus qu'une *appellation normalisée* destinée au mieux à éviter toute *tromperie sur la marchandise*. À la rigueur une norme *française* s'impose peut-être en France, une norme *européenne* n'est probablement pas impérative outre-atlantique et il se peut fort bien que ces normes locales soient en conflit avec leur contrepartie étatsunienne !

Et puis il y a un autre principe qui, souvent, s'oppose *de facto* à l'installation de nouvelles normes : la *portabilité dans le temps*. Ceci peut s'énoncer avec un soupçon de brutalité : « ce qui marchait jadis sur une installation *doit* continuer à y marcher et y donner les mêmes résultats ».

J'en ai jadis fait l'expérience il y a une trentaine d'années, alors que je dirigeais un centre de calcul. Comme à cette époque les logiciels étaient souvent « bogués » les responsables passaient du temps à inspecter les bibliothèques et à y corriger eux-mêmes les erreurs les plus flagrantes.

J'avais ainsi corrigé un sous-programme de calcul de racines carrées qui – à la suite d'une erreur de programmation en assembleur – détectait imparfaitement les tentatives d'extractions de racines carrées de réels négatifs. Le lendemain, deux chefs de service furibards débarquaient dans mon bureau : la sacro-sainte programme de dépouillement des *chambres à bulles* ne marchait plus, il se bloquait sur une affaire de racine carrée illégale !

Ayant la « norme » pour moi, je n'ai pas cédé. Mais mes interlocuteurs avaient pour eux l'argument de *portabilité* et je ne m'en suis sorti qu'en passant plusieurs jours à chercher l'erreur – imprécision de calcul d'ailleurs inoffensive – dans cet énorme progiciel, et à en pallier les effets léthaux.

## 5.2. La portabilité, c'est dans la façon d'écrire !

En fait, la pragmatique observation des « rapports de force<sup>23</sup> » fait comprendre que la solution à la portabilité... est d'écrire des T<sub>E</sub>Xtes portables ; portables non pas tellement en fonction de la *norme de droit* mais en fonction des implémentations existantes. Dans le cas de T<sub>E</sub>X (L<sup>A</sup>T<sub>E</sub>X), ceci implique de se limiter à des constructions

– acceptables avec T<sub>E</sub>X V2.x, fût-ce au prix de sacrifices esthétiques,

---

23. Qui s'exprime aussi : « on ne saute pas dans la piscine quand il n'y a pas d'eau », même quand le règlement dit qu'elle doit être pleine...

- lisibles et éditables quel que soit le codage du clavier, de l'écran et des imprimantes courantes (utilisées en dehors des pilotes T<sub>E</sub>X),
- qui produisent la même impression au propre quel que soit le jeu de polices *standard*,
- qui produisent la même impression au propre quel que soit la configuration (et l'*ordre*) des « *patterns* » de coupures de mots incluses dans le format PLAIN ou L<sup>A</sup>T<sub>E</sub>X, formats dont il est impensable de demander le changement au destinataire.

On nous objectera que ce que nous disons impensable ne l'est pas et qu'on peut changer un format en moins de quelques minutes : c'est vrai quand on est soi-même le *gourou* qui gère l'installation, mais c'est impossible quand on n'est que l'hôte d'un système multiutilisateur. Sans compter la difficulté T<sub>E</sub>Xnique pour un usager ordinaire<sup>24</sup>.

D'autre part, les T<sub>E</sub>Xtes portables doivent autant que possible se prémunir contre des changements locaux dans les styles ou certaines définitions. Certes il n'y a pas de solution générale contre tous les changements dialectaux, mais certains sont si courants qu'il vaut mieux s'en protéger.

## 6. Quelques recettes de portabilité

### 6.1. Styles et jeux de macros

La seule chose que vous êtes en droit de supposer, c'est que le destinataire possède un moteur T<sub>E</sub>X non bogué et les formats de base PLAIN et L<sup>A</sup>T<sub>E</sub>X. Un point, c'est tout. Donc :

1. Vous devez fournir le texte source de toutes les macros non incluses dans le T<sub>E</sub>Xbook[1] et tous les styles L<sup>A</sup>T<sub>E</sub>X utilisés. À la rigueur vous pouvez supposer que le partenaire possède `book.sty`, `report.sty` et `article.sty` ainsi que `A4.sty`<sup>25</sup>, `11pt.sty` et `12.sty`, mais vous courez le risque que des modifications locales y aient été apportées.

---

24. Nous avons reçu de nombreuses plaintes que MusicT<sub>E</sub>X créait des partitions incompatibles avec le papier au format *legal*, le standard américain ; alors qu'il suffisait de changer `\vsize` !

25. Attention : le format A4 n'est pas standard aux États-Unis et « plante » certaines imprimantes qui travaillent en 21×27 !

2. Les styles que vous communiquez doivent aussi être *portables*, évidemment. Ceci suppose que les auteurs de ces styles<sup>26</sup> ne s'opposent pas à leur diffusion, au moins à titre « restreint ». Dans le cas contraire on ne peut qu'affirmer une chose :

**Les textes ou programmes utilisant de façon cruciale des logiciels non portables pour cause de *copyright*... ne sont pas portables !**

## 6.2. La commutation de langues pour la coupure automatique de mots

Laissant pour plus loin le problème des règles typographiques, facilement soluble en principe par quelques macros ou un « style » L<sup>A</sup>T<sub>E</sub>X, nous considérons le problème des *coupures* automatiques de mots (termes synonymes : *division*, en anglais *hyphenation*) qui, de par l'organisation de T<sub>E</sub>X, sont figées dans les formats, qu'il est pratiquement exclu de « porter » chez les correspondants, ne serait-ce que pour des raisons d'encombrement. Alors, si on veut être bref, le plus simple est de citer Dante ALIGHIERI :

**Voi che entrate,  
lasciate ogni speranza !**

En effet, la commutation de langues est à peu près normalisée dans les 90 % d'installations de T<sub>E</sub>X de version V3.x, sous la forme `\language=\xxx`, où `\xxx` est une constante arbitraire impossible à deviner sauf courrier spécial au destinataire. Cette commutation peut être aussi activée en appelant des macros... dont l'appellation dépend fortement de la culture locale : ainsi la division à la française s'invoque selon les versions par `\fhyph`, `\french`, `\FR`, par l'affectation `\language=\l@french` ou d'autres codages connus des seuls initiés locaux. Histoire de rendre les « porteurs » de T<sub>E</sub>Xtes encore plus dépressifs, signalons aussi la forme `\language=\french` utilisée en Allemagne, où `\french` n'est plus une macro – dont l'existence peut se tester – mais... une constante.

1. Plutôt que d'invoquer une des possibilités au petit bonheur – sous prétexte que c'est elle qui marche chez vous – mieux vaut déclarer au

---

<sup>26</sup>. Ou des polices qui y sont utilisées, cas de « lucida ».

début – et avant l'ordre `\documentstyle` de  $\text{\LaTeX}$  – une séquence qui en essaye le maximum et affecte, par exemple à une macro `\l@english` ou `\l@french` les valeurs « découvertes », lesquelles seront en temps utile affectées au registre `\language`. Nous en donnons un exemple en annexe.

Tout ceci ne vous garantira pas des coupures de mots correctes, mais au moins vous aurez mis de votre côté de maximum de chances. Soyez néanmoins certains que ceci échouera dans beaucoup de cas, notamment si `\french` ou `\german` sont des constantes et non des *macros*.

2. Pour ce qui est de  $\text{\LaTeX}$ , les codages proposés (en annexe) supposent que `\french`, `\fhyph`, `\FR` et `\language=\l@french` sont des synonymes. Ce n'est souvent pas vrai, car sur certaines implémentations les commandes de langage commutent non seulement les règles de coupure de mots mais aussi *des règles typographiques* comme le retrait (*indentation*) après un titre de `\section` ou le choix typographique des guillemets ou des tirets d'`\item`. Mais ceci est un problème de portabilité des *styles* qui concerne les  $\text{\TeX}$ perts plutôt que l'écrivain ordinaire.

Notons à ce sujet que la version 3.10 de `french.sty` diffusée par l'association GUTenberg *exige* la présence d'un fichier `language.dat` décrivant – au moment de la compilation  $\text{\TeX}$  – l'ordre d'installation des règles de coupure décidée lors de la création des formats.

Si `language.dat` est inaccessible ou erroné, la commande `\french` reste indéfinie ou devient inefficace non seulement pour commuter les règles de coupure (dont `\FR` décrit ci-dessus assure un palliatif) mais aussi les règles typographiques du français. C'est donc un « style » très performant dans un environnement comptable, mais qui cumule non-portabilité, robustesse, et robustesse... dans la non-portabilité<sup>27</sup>.

Signalons cependant sa commande `\NouveauLangage`, qui reste efficace<sup>28</sup> même quand l'inexistence ou une mauvaise lecture de `language.dat` a inhibé la commande `\french`. C'est pourquoi, afin de permettre le portage du présent article entre les divers éditeurs et lecteurs, nous avons déclaré :

```
\catcode'\@=11\relax
\NouveauLangage[\l@french]{french}
\NouveauLangage[\l@english]{english}
\catcode'\@=13\relax % ou la valeur habituelle...
```

---

27. Précisons cependant que – depuis les premières moutures de cet article – l'auteur de `french.sty` prépare un *poor man french style* (`pmfrench.sty`) qui devrait être bien plus tolérant.

28. Moyennant une correction acceptée *in extremis* par l'auteur.

ce qui est un moyen de neutraliser les sécurités abusives de `french.sty` et de permettre une impression correcte presque partout où les divisions de mots à la française ont été installées dans le format L<sup>A</sup>T<sub>E</sub>X.

En fait, vu la quasi-impossibilité de détecter à coup sûr la manière de déclencher les coupures correctes d'un langage autre que l'anglais, il est plus que prudent d'en limiter l'usage dans les T<sub>E</sub>Xtes destinés à être « portés ». Il est donc sage, *hors des textes anglais*, :

- de dissuader les coupures de mots en déclarant

```
\hyphenpenalty 500\relax
```

- d'indiquer « manuellement » par `\-` les coupures possibles des mots les plus longs,
- d'user de la commande `\hyphenation{...}`.

Ensuite, faire un essai local en donnant à `\language` successivement une valeur fautive et celle de l'anglais n'est pas inutile. Et si l'on utilise `french.sty` on peut faire des essais avec un fichier `language.dat` inaccessible (le renommer) ou faux (inverser l'ordre des langues).

*REMARQUE I* : à condition d'user massivement de l'ordre `\hyphenation`, il est quand même possible d'obtenir des coupures de mots possédant des lettres accentuées tout en n'utilisant que les polices « traditionnelles » `cmxxx`. Cette possibilité – que les puristes trouveront insuffisantes mais qui a le mérite d'exister – a été décrite par D. FLIPO et L. SIEBENMANN[2]; elle consiste à introduire un fichier complémentaire `hyacc-CM.tex` qui redéfinit les macros d'accentuation et y autorise les coupures *après les lettres accentuées*, c'est-à-dire après l'ordre `\accent`.

Bien sûr, un tel dispositif ne prend vraiment effet qu'à la condition d'avoir sélectionné le mode de division de mots propre à la langue choisie mais, à défaut, l'ordre `\hyphenation` permet de pallier les plus graves incohérences.

*REMARQUE II* : Les esprits chagrins et les américanistes inconditionnels, contempteurs de tous les langages non anglo-saxons, objecteront que tout ceci est sans intérêt car les textes non anglais sont *illisibles* hors des strictes communautés linguistiques émettrices. C'est oublier que beaucoup de T<sub>E</sub>Xeurs doivent parfois travailler sur des installations dont ils ne sont pas

maîtres et dont les *system managers*<sup>29</sup> ont négligé les nécessités linguistiques non *dominantes*.

### 6.3. Les caractères de votre T<sub>E</sub>Xte

Bien plus grave que des macros inexistantes, en général facilement récupérables en répondant RETURN aux protestations de T<sub>E</sub>X, il y a le problème des caractères utilisés dans votre T<sub>E</sub>Xte. Alors là, malgré les habitudes prises par les informaticiens qui ne connaissent que le monde de « leur » ordinateur et de ceux du même modèle avec lequel ils échangent textes, programmes, commandes et progiciels exécutables, les caractères ASCII portables ne sont que ceux de code compris entre 32 (l'espace) et 126 (~). Plus, si l'on veut, le RETURN et le LINE FEED qui sont en fait effacés et reconstitués quand on transfère les fichiers en mode ASCII par disquette ou par réseau.

Il convient donc d'insister sur le fait que les *lettres accentuées*, le « ß » allemand, le « œ », le « æ », le « ø », qu'ils soient minuscules ou majuscules, ne sont portables au mieux que chez un même constructeur. On distingue en effet :

1. les normes ISO 8859 au nombre de neuf<sup>30</sup> :
  - (a) 8859-1 (LATIN1), en vigueur en France sur beaucoup de machines UNIX,
  - (b) 8859-2 (LATIN2), adapté aux langues slaves d'Europe centrale mais pas au français,
  - (c) 8859-3 (LATIN3), nécessaire au turc, à l'esperanto et au catalan,
  - (d) 8859-4 (LATIN4), nécessaire aux langues baltes,
  - (e) (LATIN5), adapté au turc et aux langues acandinaves,
  - (f) (LATIN/CYRILLIQUE),
  - (g) 8859-6 (LATIN/ARABE),
  - (h) 8859-7 (LATIN/GREC),
  - (i) (LATIN/HÉBREU),
2. le codage des Macintosh,

---

29. En français dans le texte.

30. Selon J. HENRY aux « Journées fontes » de GUTenberg en 1990.

3. le codage 850, en vigueur sur les PC,
4. d'autres codages plus exotiques sans doute hors norme,
5. et le codage 7 bits ASCII qui ignore les lettres accentuées ou nordiques.

Tout ceci fournit des arguments suffisamment péremptoires – y compris à l'intérieur d'une même communauté linguistique – pour affirmer :

1. Un T<sub>E</sub>Xte source n'est portable que s'il ne comporte que des caractères<sup>31</sup> communs à toutes les normes ISO 8859, au codage 850 et au codage des Macintosh, c'est à dire en pratique les caractères ASCII de code  $\geq 32$  et  $\leq 126$ , plus les fins de ligne qui sont gérées spécifiquement<sup>32</sup>.
2. Pour coder les lettres accentuées (et autres lettres nordiques non anglo-saxonnes) la seule solution portable est le codage donné par K<sub>N</sub>U<sub>T</sub>H dans le T<sub>E</sub>Xbook, à savoir `\'e`, `\ss{}`, `\^i{}`, `\ae{}`, etc.
3. Il faut éviter au maximum les enregistrements de plus de 80 caractères significatifs, car ils risquent toujours d'être coupés ou tronqués par les messageries ou... les longueurs fixes d'enregistrements sur des systèmes VM ou MVS.

Bien sûr, pour que ces lettres accentuées ne bloquent pas les coupures, il est indispensable que `\'e` ne soit pas traduit par `{\accent 19 e}` mais

- soit par un caractère de `\catcode 11` ou `12`, en l'occurrence `^^e9`, c'est-à-dire le « é » des polices `dmxxx` ou `dcxxx`, ce qui revient alors à supposer (ou à savoir) que l'implémentation destinataire est bien un T<sub>E</sub>X V3.x capable de gérer de manière interne des codes de caractères à 8 bits et disposant des polices `dmxxx` ou `dcxxx`,
- soit par les modifications incluses dans `hyacc-CM.tex` qui autorisent[2] les coupures *après les lettres accentuées* par l'ordre `\accent`.

---

31. Caractères strictement informatiques, ce qui n'a que peu à voir avec les caractères typographiques finaux.

32. Et je me suis fait vertement tancer pour avoir diffusé des exemples d'usage de MusicT<sub>E</sub>X contenant des « paroles » françaises avec lettres accentuées au code 850 (codage PC) qui, chez mes correspondants, devenaient des « *illegal characters* ».

Nous verrons plus loin (6.5) comment savoir – dans T<sub>E</sub>X – quel est le type de polices ayant cours mais, dans tous les cas de figure, les macros redéfinissant ainsi les accentuations doivent être mises dans un fichier séparé ou un style personnel, et un message explicite doit inciter le destinataire à omettre ce style ou l'ordre `\input` en cas de problème.

Indépendamment du recodage des macros d'accentuation se pose le problème de la transformation de toutes les lettres accentuées d'un T<sub>E</sub>Xte pour revenir au codage de K<sub>N</sub>U<sub>T</sub>H. Plusieurs solutions sont envisageables :

- un coup d'éditeur (commande `query-replace-string` d'EMACS, de JOVE ou l'équivalent en *Turbo-Pascal* ou *Turbo C*). C'est facile, mais il faut le faire pour tous les caractères accentués possibles (une douzaine pour le français, deux fois plus si on accentue des majuscules) ;
- sur PC (sans doute aussi d'autres) l'utilitaire `texconv` distribué avec EM<sub>T</sub>E<sub>X</sub> fait ça en une seule commande :

```
texconv -a 850_tex.tcp <source.tex> <portable.tex>
```

L'équivalent pour Macintosh existe et s'appelle **Multi-Convert**<sup>33</sup>.

- Bernd RAICHLE a également développé un petit utilitaire, écrit en T<sub>E</sub>X, qui fait cette même conversion en se basant sur un fichier `convert.tbl` – qu'il faut donc adapter au codage de son propre ensemble [clavier + ordinateur + T<sub>E</sub>X]<sup>34</sup> – que l'on trouve sur le serveur `ftp.uni-stuttgart.de`<sup>35</sup>. Étant écrit en T<sub>E</sub>X, il tourne sur n'importe quel ordinateur et on l'appelle par la commande `tex convert`, en répondant simplement aux questions qu'il pose.
- enfin, si vous n'avez pas ces utilitaires, leur écriture en Fortran, en Pascal ou en C est à la portée de n'importe quel programmeur débutant...

*REMARQUE* : Beaucoup de T<sub>E</sub>Xeurs émérites sont en total désaccord avec ce qui précède, considérant que ISO 8859-1 (LATIN1) n'est pas un « codage » quelconque mais « la norme » et que tous les ordinateurs qui ne la respectent pas, soit par absence – ce qui est le moins grave – comme les claviers et consoles

---

33. Peut être obtenu par `ftp anonymous` sur `matups.matups.fr`.

34. Se méfier du gag des conversions de *code page* qui, si elles sont actives, changent déjà les caractères avant de les soumettre à T<sub>E</sub>X!

35. Numéro IP : 129.69.1.12., *directory* `soft/tex/utilities/convert`.

QWERTY, soit par déviance comme les Macintosh ou les PC européens utilisant le codage 850, sont en *faute*.

Vous êtes donc en *droit* d'exiger que vos correspondants soient capables d'accepter un T<sub>E</sub>Xte codé ISO 8859-1 « LATIN1 », et de considérer comme négligeables tous ceux qui en seraient incapables.

Cela étant dit, l'avis *strictement personnel* – qui n'engage surtout pas l'association GUTenberg – de l'auteur est qu'il est prudent de ne pas aller jusqu'au bout de son bon droit et d'accepter des sacrifices, si l'on veut vraiment être lu.

#### 6.4. La sélection des polices

Au début, la police du texte maigre normal avec empattement (*serif*) s'appelait **amr10** ; elle a été remplacée il y a plusieurs années par **cmr10**. Puis, afin de permettre la coupure des mots accentués on a créé les polices « virtuelles », de sorte que l'on invoque souvent<sup>36</sup> **dmr10**. Bien plus récemment a été introduite la police accentuée « réelle » **dcr10** qui se substitue de plus en plus à **dmr10**, notamment en Europe centrale (tchèque, polonais, slovène, etc.) en attendant la police « européenne » définitive qui sera **ecr10**.

Le plus souvent, pour invoquer la police droite, maigre et avec empattement en 10 points, il suffit de dire `\rm` ou – pour s'affranchir en L<sup>A</sup>T<sub>E</sub>X du corps général `11pt` ou `12pt` mentionné dans `\documentstyle[...]` – de dire plus précisément `\tenrm`. Il y a cependant au moins deux cas où cette référence aux définitions du format PLAIN ou L<sup>A</sup>T<sub>E</sub>X est insuffisantes ou donne des choses non portables :

1. Quand on veut une police particulière avec un corps sortant de l'ordinaire ; ainsi, pour avoir du gras sans empattement (*sans serif*) en corps de 24 points (approximativement), doit on déclarer :

```
– \font\enorme=amssbx10 scaled \magstep4,
– \font\enorme=cmssbx10 scaled \magstep4,
– \font\enorme=dmssbx10 scaled \magstep4,
– \font\enorme=dcssbx10 scaled \magstep4,
– ou \font\enorme=ecssbx10 scaled \magstep4 ?
```

---

36. Même si l'impression finale utilise la police **cmr10** en y superposant lettres et accents pour simuler les lettres accentuées de **dmr10**.

2. Quand on veut dans une formule mathématique mettre un « lambda majuscule gras » on pouvait habituellement coder `\bf\Lambda` tant que les polices courantes étaient `amxxx`, `cmxxx` ou `dmxxx`. Malheureusement ceci ne marche plus avec les polices `dcxxx` pour la raison suivante :
  - les lettres grecques majuscules – qui se trouvaient dans les polices `cmxxx` ont été éliminées des polices `dcxxx` au profit d’un jeu complet d’accents... et de « guillemets à la française » ;
  - les formats des polices `dcxxx` ont créé une nouvelle famille (`\fam15`), invoquant par exception les anciennes polices `cmxxx` pour y récupérer les caractères disparus des nouvelles polices `dcxxx`, comme les majuscules grecques ;
  - Alors que `\Lambda` invoque le caractère numéro 03 d’une police `cmr10` (grandie le cas échéant), la commande `\bf` invoque – si l’on travaille avec les polices `dcxxx` – la police `dcbx10` qui n’a plus la lettre grecque là où l’avait la police `cmbx10` ;
  - Le pire est qu’il n’y a aucun diagnostic et qu’on ne découvre l’erreur que lors d’un examen attentif des épreuves.

Pour toutes ces raisons, les contraintes de portabilité interdisent :

1. de déclarer `\font\xxx=` autre chose qu’une police `cmxxx` car ce sont les seules dont on soit certain qu’elles existent partout<sup>37</sup>. À moins bien sûr que ce soit une police spécifique que l’on a communiquée spécialement au correspondant distant<sup>38</sup>.
2. de ne jamais mettre de lettres accentuées, quel qu’en soit le codage, dans le texte gouverné par une police explicitement déclarée `\font\xxx=cm...`, car on ne sait pas si, par exemple, le code `\’e` va se trouver transcrit par les macros du site destinataire sous la forme `{\accent 19 e}` ou sous la forme `^^e9` qui invoque un caractère inexistant dans la police `\xxx` ci-dessus déclarée. Ou alors, si l’on tient à accentuer une lettre, il faudra sacrifier les possibilités de coupure automatique et coder sous la forme *explicite* : `{\accent 19 e}`<sup>39</sup>.

---

37. En espérant ne pas tomber sur une installation utilisant encore et uniquement les polices `amxxx` !

38. Ce qui suppose que l’usager ordinaire – notamment celui d’un *mainframe* sait et a le droit d’installer des polices en plus de celles qui y sont *standard*.

39. Cette valeur de l’accent aigu est valable pour les polices `cmxxx` et `dmxxx`, mais elle est de 01 si les polices sont de la série `dcxxx` ou `ecxxx`, rappelons-le!

3. de spécifier des changements de « graisse » ou d'inclinaison (c'est-à-dire `\rm`, `\sl`, `\it`, `\bf`, etc.) dans le mode mathématique : ça imprime des choses gravement fausses avec les polices `dcxxx` et notamment avec Euro-OzT<sub>E</sub>X<sup>40</sup>.

### 6.5. Pour les T<sub>E</sub>Xperts en portabilité : tester les noms de polices

Le problème de la portabilité n'est pas spécifique de T<sub>E</sub>X ; bien avant, les auteurs de logiciels ont été affrontés aux problèmes de faire marcher des programmes dans divers environnements, sans se limiter à l'intersection – parfois vide – des possibilités offertes par les divers systèmes. Ceci implique que les logiciels portables puissent tester les possibilités du système dans lequel ils tournent et effectuent *conditionnellement* les opérations adaptées à chaque « environnement ».

Ces tests, les « enquêtes d'environnement » existent dans les langages de programmation depuis une vingtaine d'années : ALGOL 68, puis ADA, puis Fortran 90 l'offrent ou l'ont offert explicitement. Et même en Fortran 77, l'auteur de ces lignes a fréquemment testé la valeur de `CHAR(' ')` pour savoir si son programme s'exécutait dans le monde EBCDIC ou ASCII afin de traiter correctement des problèmes de majuscules, de minuscules ou d'ordre alphabétique.

T<sub>E</sub>X lui même offre explicitement beaucoup de possibilités d'« enquêtes d'environnement », mais peu de choses sont dites sur le moyen de savoir si un T<sub>E</sub>Xte est compilé dans un environnement de polices `cmxxx`, `dmxxx`, `dcxxx` ou autre. Ce moyen existe pourtant, et il est simple et décrit dans le T<sub>E</sub>Xbook[1] (chapitre 20, p. 213) mais nous remercions cependant Bernd RAICHLE qui nous a orienté dans la direction *ad hoc*. Voici donc cette solution, dont on peut évidemment changer les noms de macros :

```
\def\wsplita#1#2#3.XX.#{1}%
\def\wsplitb#1#2#3.XX.#{2}%
\rm
\edef\fonthdg{\expandafter\wsplita\fontname\the\font.XX.}%
\edef\fonthdge{\expandafter\wsplitb\fontname\the\font.XX.}%
```

Après l'exécution de ce petit code T<sub>E</sub>X, `\fonthdg` vaut le premier caractère du nom de la police invoquée par `\rm`, et `\fonthdge` vaut son

---

40. Vrai fin 1992...

deuxième caractère. Alors, sachant les règles de constitution des noms de polices, on peut invoquer la police *sans sérif* grasse de corps 12 de la même série que `\rm` en codant :

```
\font\twelvssbx=\fonthdg\fonthdgs sbx12
```

ce qui donnera `cmssbx12` si `\rm` valait `cmr10`, ou `dcssbx12` si `\rm` valait `dcr10`.

Ensuite, on peut par exemple coder :

```
\immediate\write16{ Polices utilisees:  
                  \fonthdg\fonthdgs}\relax  
\newif\ifeurotex  
\if\fonthdg d\relax  
  \if\fonthdgs c\relax \eurotextrue  
  \fi  
\fi
```

qui imprimera le préfixe des polices utilisées et qui rendra « vrai » le test `\ifeurotex` si ce préfixe est `dc`. Bien sûr, `\ifeurotex` pourra être judicieusement testé pour savoir quelle police et quel `\char` invoquer pour avoir – par exemple – un *lambda majuscule gras* extrait des polices `cmbx...`, ou sélectionner `\accent 19` ou `\accent "01` pour mettre un accent aigu sur un caractère quelconque.

Ceci étant exposé, un certain nombre de lecteurs attentifs et convaincus vont se dire « il a raison, je vais mettre la déclaration de `\fonthdg` et de `\fonthdgs` » directement dans mes formats `PLAIN` et `LATEX`. Certes cela ne mange pas de pain, mais la place d'un tel test n'est pas dans le format – que vous ne transmettez pas à votre correspondant lointain – mais *dans le texte portable* !

*REMARQUE* : les exemples donnés ici sont limité aux cas où les polices ont des noms dont les deux premiers caractères permettent de connaître la carte des codes. Avec `TimesLATEX`, il faudrait faire des tests plus compliqués.

## 7. La portabilité des textes en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

### 7.1. La portabilité des styles

Sachant que ces styles doivent un jour ou l'autre être transportés pour permettre la compilation de textes principaux qu'ils assaisonnent, leur rédaction doit obéir aux mêmes règles que le  $\text{T}_{\text{E}}\text{X}$ te portable, à savoir :

1. rien que des caractères ISO 8859  $\geq 32$  et  $\leq 126$  (c'est-à-dire ASCII),
2. pour les déclarations de polices (`\font\xxx=`):
  - soit des polices de série explicite `cmxxx`,
  - soit des noms de polices préfixés comme indiqué en 6.5,
  - soit des polices spéciales fournies avec le texte,
3. aucune accentuation des caractères des polices introduites par leur nom explicite `cmxxx`.

Là où la situation se complique, c'est lorsque ces styles ont une composante linguistique : `french.sty`, `german.sty`, `english.sty`, etc. Ces styles linguistiques mélangent alors souvent :

- les commandes de présentation : retraits de paragraphes, présentation des guillemets, règles typographiques « nationales », etc.,
- les commandes de commutation de `\language` pour une coupure correcte des mots.

Or, si les commandes du premier groupe sont des macros indépendantes du format  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  installé, la valeur entière que l'on doit affecter à `\language` pour sélectionner la coupure à la française, à l'anglaise ou à l'allemande est *imposée* par les choix faits lors de la création du format  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  (et aussi PLAIN). Aussi, un style ou un  $\text{T}_{\text{E}}\text{X}$ te qui impose la commande `\language=1` pour la coupure française n'est absolument pas portable, car l'autre implémenteur a pu choisir `\language=1` pour le néerlandais et `\language=2` pour le français.

On pourrait donc dire qu'il ne faut pas utiliser ces styles `french.sty`, `german.sty`, `english.sty`, etc., du fait de leur non portabilité (actuelle), mais ce serait se priver de nombreuses autres facilités qu'ils offrent pour

des conventions de mise en page, les guillemets ou les titres standard de rubriques comme « chapitre », « résumé », « table des matières » et leurs équivalents en allemand ou en italien. Nous ne pouvons hélas pas donner de solution générale mais, le plus souvent, une lecture attentive des notices permet de trouver des commandes « de secours », indépendantes de la valeur à attribuer à `\language`, qui réalisent les fonctions recherchées.

## 7.2. Les « dirty tricks » de NFSS

NFSS<sup>41</sup> est une variante performante du mode de sélection des polices par  $\text{\LaTeX}$ , dont la principale caractéristique est d’avoir dissocié les paramètres définissant les polices en quatre classes :

1. la famille (*family*): roman (avec empattement), *sans sérif*, télétype et *fibonacci*<sup>42</sup> ;
2. la série (*series*): *medium size* (exemple `cmr10`, `cmss12`), *bold extended* (exemple `cmbx10`, `cmssbx12`), etc.;
3. la forme (*shape*): droit, italique, *slanted*, petites capitales, etc.;
4. l’alphabet (NFSS, version 2).

Ceci facilite considérablement l’usage des polices moins courantes comme *sans sérif gras* en déclarant simplement `\bf\ss` ou `\ss\bf` ou bien *penché gras* en déclarant simplement `\bf\sl` ou `\sl\bf`.

Mais l’inconvénient de NFSS, quand il est incorporé par défaut dans le format appelé simplement  $\text{\LaTeX}$ , est d’avoir offert ces possibilités en *modifiant* l’effet des commandes *standard* de changement police : `\bf`, `\tt`, `\sl`, etc. Souvent les commandes modifiées donnent les mêmes résultats, mais il faut connaître de nouvelles commandes pour revenir du *gras droit avec empattement* au *maigre penché (slanted)*, car `\sl` et `\bf` qui étaient des commandes « exclusives » avec le  $\text{\LaTeX}$  traditionnel deviennent « orthogonales » avec NFSS.

Ceci serait sans aucune gravité si NFSS était invoqué par un *style*  $\text{\LaTeX}$  qu’il suffirait de transmettre au correspondant si nécessaire<sup>43</sup> ;

---

41. *New Font Selection Scheme*, écrit par Frank MITTELBACH et Rainer SCHÖPF.

42. Polices `cmfi10`, `cmfib8`, etc.

43. Notons qu’il n’est pas difficile de prendre les fichiers spécifiques de NFSS pour en faire une « option de style ».

malheureusement NFSS est installé comme *norme* dans bon nombre d'installations de L<sup>A</sup>T<sub>E</sub>X<sup>44</sup> qui de ce fait ont des comportements différents des L<sup>A</sup>T<sub>E</sub>X classiques.

Tout ceci étant dûment récriminé, c'est au rédacteur d'un L<sup>A</sup>T<sub>E</sub>Xte de se prémunir lui-même contre les mauvaises plaisanteries d'un NFSS peut-être installé chez lui ou chez le destinataire, sans qu'il en soit conscient. Il faut donc coder les changements de police d'une manière qui donne les mêmes résultats, que NFSS soit actif par défaut ou non<sup>45</sup>. La solution simple consiste à déclarer :

```
\def\resetNFSS{\csname normalshape\endcsname
\csname mediumseries\endcsname \rm}
```

et à faire précéder tout changement de police par `\resetNFSS` que l'on invoque aussi par sécurité<sup>46</sup> juste après le `\begin{document}`.

## 8. Suppliques aux auteurs de styles et aux aménageurs de formats

Nous avons exposé dans ce qui précède un certain nombre de chaussetrapes qui guettent les auteurs qui souhaitent transmettre des T<sub>E</sub>Xtes corrects hors frontières ou même hors de leur simple domaine d'influence, et nous avons indiqué quelques embryons de recettes permettant *au moins mal* de se sortir d'affaire. Au passage, nous avons dû égratigner d'honorables auteurs de logiciels pour quelques erreurs stratégiques qui, d'ailleurs, n'enlèvent rien à la qualité générale de leur œuvre.

Disons tout de suite qu'il nous paraît hors de question de demander que ces *erreurs stratégiques*<sup>47</sup> soient corrigées : en effet, dès lors que ces logiciels ont été diffusés, il est certain qu'il en restera quelque part dans le monde des versions anciennes, et la modification d'un logiciel déjà largement diffusé ne ferait souvent qu'accroître la confusion et les problèmes de portabilité. Donc

---

44. Dont Euro-OzT<sub>E</sub>X, sans mise en garde décelable dans la notice!

45. La documentation NFSS suggère de mettre `oldfonts` dans les options de style mais que se passe-t-il si on invoque l'option de style `oldfonts` alors que le fichier `oldfonts.sty` n'existe pas?

46. On remarquera que `\csname normalshape\endcsname` exécute la procédure NFSS `\normalshape` si elle existe et `\relax` si elle n'existe pas.

47. Ce que nous qualifions d'*erreur* peut aussi être une volonté d'imposer *urbi et orbi* des normes jusqu'à présent locales, qui sont *par définition* les meilleures...

les suppliques qui suivent s'adressent plutôt aux auteurs de futurs styles, logiciels ou formats.

### 8.1. Un impératif: la compatibilité avec le passé

- Sauf s'il s'agit d'une « bogue » dont personne ne s'accommode et que tout le monde évite ou contourne, toutes les commandes d'une *nouvelle version* d'un format, d'un moteur ou d'un style doivent avoir le même effet<sup>48</sup> que dans les précédentes versions. On doit en particulier s'interdire de changer *dans les formats de base* les effets des commandes décrites dans le `TEXbook[1]` ou dans la définition de `LATEX[3]`<sup>49</sup>.
- Si l'on veut changer le comportement des commandes *standard* il faut soit leur donner un nouveau nom, soit donner un nouveau nom au style ou au format... et toujours laisser – sous l'ancien nom – l'ancien style ou l'ancien format accessible<sup>50</sup>.

### 8.2. La commutation des modes de coupure de mots

On l'a vu précédemment, toute la difficulté est de détecter la valeur à attribuer au registre `\language` pour la langue souhaitée.

On a cru – en Europe – résoudre le problème en ayant un fichier pseudo-standard `language.dat`<sup>51</sup> qui déclare les noms des commandes de contrôle de coupure (`\english`, `\french`, `\german`, etc.) dans l'ordre croissant des valeurs à donner au registre `\language`, et en s'y référant *à la fois* lors de la création des formats PLAIN et `LATEX`, et lors de la lecture des styles linguistiques comme `french.sty`. Le malheur est que cette façon de procéder – intrinsèquement satisfaisante<sup>52</sup> – n'est appliquée que dans les distributions de style `french` par l'association GUTenberg, et qu'elle est ignorée par `EMTEX` et même dans la distribution Euro-Oz`TEX` !

Pour sa part, la distribution `EMTEX` se heurte à la même difficulté d'identification des modes de coupures mais, au lieu de transmettre l'ordre

---

48. Au temps de calcul et à l'encombrement près.

49. Cité[4] par Michael URBAN/Éric CORNELIS auquel nous nous référons usuellement.

50. Si par exemple NFSS avait appelé `\Rm`, `\Bf`, `\S1` les nouvelles commandes en redéfinissant `\rm`, `\bf` et `\s1` comme combinaison des nouvelles, ou bien si l'ensemble `LATEX+NFSS` s'était appelé `LATEX_NFSS`, il n'y aurait eu aucun problème. Mais maintenant il est trop tard pour revenir en arrière.

51. Nom de fichier constant et non paramétrable.

52. Sauf si, pour des contraintes de place on doit avoir plusieurs formats de base avec des groupements de langues différents, ce qui est incompatible avec l'unicité de `language.dat`.

d'installation des « *hyphenation patterns* » par un fichier `language.dat` elle suppose que des valeurs `\l@english`, `\l@USenglish`, `\l@german`, `\l@austrian` et `\l@french` ont pu être définies quelquepart dans le format PLAIN ou L<sup>A</sup>T<sub>E</sub>X, et qu'il suffit pour couper les mots à la mode française de dire

```
\language=\l@french
```

Communiquer ainsi les numéros des divers langages (de leurs *patterns* de division) est infiniment plus portable que de postuler l'existence d'un fichier `language.dat`, car alors l'information est véhiculée *dans chaque format* alors que rien ne prouve que le `language.dat` que le moteur T<sub>E</sub>X va trouver<sup>53</sup> soit justement celui qui a servi à constituer le format que l'on utilise. Notons avec satisfaction que les plus récentes distributions du fichier `hyconfig.tex`<sup>54</sup> fabriquent les macros `\l@french`, `\l@english`, `\l@german`, etc., à l'occasion de la lecture de `language.dat` lors de la création des formats de base PLAIN et L<sup>A</sup>T<sub>E</sub>X.

Malgré ce progrès, il reste toujours impossible de savoir de façon *standard*, donc quelle que soit l'implémentation destinataire, quelle valeur on doit donner à `\language` pour couper les mots à la mode française, anglaise, allemande ou italienne. C'est pourquoi nous recommandons que les « styles » à caractère linguistique soient coupés en deux parties :

1. une partie strictement portable où les références aux *patterns* de coupures utilisent exclusivement les conventions déjà communes<sup>55</sup> aux distributions GUTenberg et E<sup>M</sup>T<sub>E</sub>X, soit :

```
\language=\l@english
\language=\l@USenglish
\language=\l@french
\language=\l@german
\language=\l@austrian
\language=\l@...
```

---

53. S'il le trouve, ce qui n'est justement pas le cas avec OzT<sub>E</sub>X 1.4.1 ni PD VMS 3.3a.

54. Adjonction datée du 24 août 1992.

55. Sans doute tirées de Babel-T<sub>E</sub>X.

2. une partie aussi robuste que possible mais non strictement portable – et qu'on ait par conséquent le droit de modifier sans verrouillage par le *copyright* – pour « trouver » quelle valeur il convient localement de donner aux divers `\l@<language>`, s'ils ne sont pas définis.

Cette partie robuste et ajustable pourrait s'inspirer de la séquence indiquée en annexe pour deviner les règles locales de commutation de langues, complétée éventuellement par un décryptage du fichier `language.dat`.

### 8.3. Les macros d'accentuation

À l'exception de `CaesarCM[5]`, toutes celles que nous avons trouvées supposent que la police active est dans la série « standard » de l'implémentation. Ainsi, sur un format basé sur les polices `cmxxx`, la macro `\'` invoquera `\accent 19` même si la police active est intentionnellement de la série `dmxxx` – ce qui nuit aux coupures automatiques – et même si la police active est `dcr10`, auquel cas l'impression sera *fausse*.

C'est pourquoi nous estimons que ces macros d'accentuation (y compris les caractères « actifs » issus des claviers européens) devraient<sup>56</sup> :

1. tester le nom de la police courante, ou au moins ses deux premiers caractères,
2. tester si le caractère accentué résultant est une lettre unique de cette police – sachant que le résultat du test dépend à la fois de la police et du caractère à accentuer<sup>57</sup> – auquel cas on invoque le caractère résultant sous la forme `^^xx`,
3. si aucun caractère unique n'est disponible dans la police courante, alors recourir à la primitive `\accent`, sachant que le numéro d'accent dépend lui aussi du nom de la police courante.
4. traiter en plus le cas particulier où la lettre à accentuer est `\i`, dont le codage n'est hélas pas le même pour les polices `cmxxx/dmxxx` et `dcxxx...`

Notons qu'une telle redéfinition robuste et portable peut fort bien se faire dans un fichier à inclure par `\input` ou dans une option de style `LATEX`.

---

56. C'est semble-t-il ce que fait `caesarCM[5]`.

57. Ainsi le `é` existe dans les polices `dcxxx`, mais pas dans `dmxxx`, alors que `é` existe dans les deux.

## 9. Conclusions

De trois choses l'une :

1. ou bien vous avez réussi à lire et comprendre tout ce qui précède, ce qui signifie que vous êtes déjà un puissant T<sub>E</sub>Xpert ; dans ce cas l'auteur vous prie de l'excuser de vous avoir fait perdre votre temps à le lire ;
2. ou bien – malgré notre intention d'être le plus concis possible – vous retirez l'impression que la portabilité de T<sub>E</sub>X est une affaire démentiellement impraticable ; dans ce cas nous rappellerons qu'un correspondant de la liste GUT@FRULM11 s'est récemment offusqué d'avoir entendu dire en des instances réputées compétentes : « *tout le monde sait que T<sub>E</sub>X ne sait pas couper les mots avec lettres accentuées* » ; eh bien, l'interlocuteur cité par le correspondant n'avait que très partiellement tort, car en fait :
  - seuls les T<sub>E</sub>Xperts savent configurer et écrire du T<sub>E</sub>X où les mots accentués qui soient portablement... coupables<sup>58</sup> !
  - les seuls T<sub>E</sub>Xtes « proprement » portables sont soit en anglais, soit avec indication explicite (\-) ou verrouillage des possibilités de coupure de mots (commande `\hyphenpenalty= beaucoup`) ;
3. ou bien, vous n'avez pas tout compris mais vous voulez un résumé des « consignes » de portabilité. Alors lisez cette dernière section :

## 10. Neuf un conseils pour la portabilité des T<sub>E</sub>Xtes

1. fournir toutes vos macros et tous vos styles L<sup>A</sup>T<sub>E</sub>X, à l'exception des standards internationaux (`\report`, `\article`, etc.) ;
2. n'utiliser que des *styles* portables<sup>59</sup> ;
3. tester – au moins – les deux premières lettres du nom des polices actives et y adapter les macros d'accentuation<sup>60</sup> ;

---

58. « Coupable » signifie : « que l'on peut couper », mais qu'alliez-vous donc penser ?

59. Ce qui exclut `fr french.sty` de GUTenberg, au moins jusqu'à la version 3.10 comprise et probablement `german.sty` d'EMT<sub>E</sub>X.

60. Ceci ne marche pas avec le polices *postscript*... mais toutes les imprimantes ne sont pas *postscript*, donc ces polices ne sont pas portables.

4. récupérer ces deux premières lettres pour préfixer toute police déclarée par `\font` ;
5. n'utiliser ni `\bf`, ni `\sl`, ni `\it` en mode mathématique ;
6. pour ce qui est de NFSS, ou bien :
  - ne pas utiliser NFSS<sup>61</sup> ;
  - mieux encore : imprimer votre texte *avec* et *sans* NFSS ; les deux résultats doivent être identiques ;
  - utiliser NFSS en en faisant une « option de style »... et expédier cette option avec votre  $\text{\LaTeX}$ te.
7. déployer des trésors d'ingéniosité pour écrire des macros qui trouvent les numéros de `\language` dans une installation destinataire... que vous ne pouvez probablement pas tester directement !
8. augmenter `\hyphenpenalty` pour dissuader les coupures de mot hors de l'anglais, ou les spécifier explicitement par `\-`.
9. n'utiliser que le sous-ensemble 7 bits de la norme ISO 8859 (caractères ASCII)<sup>62</sup> 7 bits  $\geq 32$  ;

## Références bibliographiques

- [1] Donald E. KNUTH, « The  $\text{\TeX}$ book », *Addison-Wesley Publishing Company*, 1984.
- [2] Daniel FLIPO, Laurent SIEBENMANN, « Hyphenation in presence of accents and diacritics », *Proceedings of the 7th European  $\text{\TeX}$  Conference, Czechoslovak  $\text{\TeX}$  Users Group*, Septembre 1992, pp. 87–96.
- [3] Leslie LAMPORT, «  $\text{\LaTeX}$  A document Preparation System », *Addison-Wesley Publishing Company*, 1986.
- [4] Michael URBAN/Éric CORNELIS, « Premiers pas en  $\text{\LaTeX}$  », *Cahier GUTenberg*, 1991.
- [5] Laurent SIEBENMANN, « CaesarCM, A gentle road to perfect hyphenation in modest  $\text{\TeX}$  », *Proceedings of the 7th European  $\text{\TeX}$  Conference, Czechoslovak  $\text{\TeX}$  Users Group*, Septembre 1992, pp. 97–106.

---

61. Le neutraliser sous Euro-Oz $\text{\TeX}$  avec l'option de style `oldfonts`.

62. Avis personnel car vous êtes en droit d'exiger que vos correspondants acceptent le codage normalisé ISO 8859-1 (LATIN1)... et de traiter par le mépris ceux qui n'y arrivent pas. Notons qu'au même titre, des correspondants lithuaniens sont alors en droit d'exiger de vous que vous traitiez correctement ISO 8859-4 (LATIN4) et des grecs de vous faire accepter ISO 8859-7.

## Annexe : à la découverte des modes de coupure des mots selon les langues

```

\catcode'\@=11\relax
% palliatif a une bogue de jadis dans hyconfig.tex
\ifx\workl@\undefined\relax \else\workl@=999\relax
\fi
\ifx\l@french\undefined
  \language=999\relax
  \csname FR\endcsname
  \csname fhyph\endcsname
  \csname french\endcsname
  \ifnum\language < 100\relax\edef\l@french{\the\language}%
  \fi
\fi
\ifx\l@english\undefined
  \language=999\relax
  \ifx\GB\undefined % Euro-0zTeX prévoit les deux !
    \csname US\endcsname
  \else\GB\fi
  \csname ehyp\endcsname
  \csname english\endcsname
  \ifnum\language < 100\relax\edef\l@english{\the\language}%
  \fi
\fi

\ifx\l@french\undefined
  \def\FR{\message {Sorry, no French hyphenation !}}%
\else\def\FR{\language=\l@french\relax}
\fi
\ifx\l@english\undefined
  \def\US{\message {Sorry, no English hyphenation !}}%
\else\def\US{\language=\l@english\relax}
\fi
\catcode'\@=13\relax

```

Même chose pour les autres langues éventuelles. Ensuite vous mettez la commande `\FR` pour passer au mode de coupure français et `\US` pour le mode de coupure anglosaxon.

Vous pouvez aussi vous prémunir contre une implantation vétuste de T<sub>E</sub>X V2.x qui ne connaisse pas le registre `\language`, peut-être explicitement invoqué dans votre texte ou dans vos styles L<sup>A</sup>T<sub>E</sub>X, en déclarant<sup>63</sup> aussi :

```

\newcount\dummylanguage
\ifx\language\undefined \def\language{\dummylanguage}\fi

```

---

63. C'est ce que fait en substance `french.sty` version GUTenberg 3.10, mais une telle précaution mérite d'être prise même si l'on n'utilise pas `french.sty`.