
Liste des péchés des utilisateurs de L^AT_EX 2_ε

ou

Commandes et extensions obsolètes, et autres erreurs

Marc Ensenbach, Mark Trettin
courriel : Marc.Ensenbach@post.rwth-aachen.de

Version 2.3 du 20 septembre 2011

Traduction française de Bernard Alfonsi
courriel : bernard.alfonsi@free.fr

Résumé

Ceci est la version française de « l2tabu ». — La rédaction de ce bref aperçu a été suscitée par une discussion dans le newsgroup T_EX de langue allemande^a, où revenaient de façon récurrente des extensions et des commandes obsolètes ou « mauvaises ». Nous essayons dans cet article de montrer les erreurs les plus communes et de proposer des solutions différentes. Cet article n'entend pas remplacer des introductions comme l2kurz [11] ni De-TeX-FAQ^b, ni The UK T_EX FAQ [3], mais offrir un simple aperçu. Nous serions reconnaissants de vos suggestions, améliorations et commentaires.

a. de.comp.text.tex

b. <http://projekte.dante.de/DanteFAQ/WebHome>

Copyright © 2003–2007 by Mark Trettin, 2009–2011 by Marc Ensenbach.

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Cette documentation peut être distribuée seulement aux termes et conditions de l'*Open Publication License*, v1.0 ou postérieure (la version la plus récente est actuellement disponible à l'adresse : <http://www.opencontent.org/openpub/>).

Les noms de *software* et de *hardware* mentionnés dans cet article sont pour la plupart, des marques déposées, soumises, comme telles, aux dispositions légales.

Nous remercions :

Ralf Angeli, Christoph Bier, Christian Faulhammer, Jürgen Fenn¹, Ulrike Fischer, Ralf Heckmann, Yvon Henel², Yvonne Hoffmüller, David Kastrup, Markus Kohm, Thomas Lotze, Gonzalo Medina Arellano³, Frank Mittelbach, Heiko Oberdiek, Mauro Sacchetto⁴, Walter Schmidt, Uwe Siart, Stefan Stoll, Knut Wenzig, Emanuele Zannarini⁵ et Reinhard Zierke,

pour leurs conseils, remarques et corrections. Au cas où nous aurions oublié quelqu'un, qu'il veuille bien nous envoyer un courriel.

Traduction française à partir de la version 2.3, de sa traduction italienne et de la traduction française de la version 1.7 faite par Le T_EXnicien de surface.

-
1. Traduction anglaise : <CTAN:info/l2tabu/english/l2tabuen.pdf>
 2. Traduction française : <CTAN:info/l2tabu/french/l2tabufr.pdf>
 3. Traduction espagnole : <CTAN:info/l2tabu/spanish/l2tabues.pdf>
 4. Traduction italienne : <CTAN:info/l2tabu/italian/l2tabuit.pdf>
 5. Traducteur de la première version italienne, 2005

Table des matières

1. « Péchés mortels »	1
1.1. a4.sty, a4wide.sty	1
1.2. Modifier la mise en page	1
1.3. Changer les extensions et les classes	1
1.4. Modification de l’interligne avec \baselinestretch	2
1.5. Indentation et espace entre les paragraphes – \parindent, \parskip . .	2
1.6. Mettre en évidence les formules mathématiques avec \$\$	3
1.7. \def & \newcommand	4
1.8. Utilisation de \sloppy	4
2. Commandes, classes et extensions obsolètes	6
2.1. Commandes	6
2.1.1. Modification du style de caractères	6
2.1.2. Fractions mathématiques (\over & \frac)	7
2.1.3. Centrer du texte avec \centerline	7
2.2. Fichiers de classes et d’extensions	8
2.2.1. scrlettr.cls & scrlettr2.cls	8
2.2.2. epsf.sty, psfig.sty, epsfig.sty & graphics.sty, graphicx.sty	8
2.2.3. doublespace.sty & setspace.sty	9
2.2.4. fancyheadings.sty, scrpage.sty & fancyhdr.sty, scrpage2.sty . . .	9
2.2.5. La famille d’extensions caption.sty	9
2.2.6. isolatin.sty, umlaut.sty & inputenc.sty	10
2.2.7. t1enc.sty & fontenc.sty	11
2.2.8. natdin.bst & dinat.bst	12
2.2.9. glossary.sty & glossaries.sty	12
2.2.10. SIstyle.sty, SIunits.sty & siunitx.sty	12
2.2.11. subfigure.sty, subfig.sty & subcaption.sty	12
2.3. Caractères	13
2.3.1. times.sty	13
2.3.2. mathptm.sty	14
2.3.3. pslatex.sty	14
2.3.4. palatino.sty	14
2.3.5. mathpple.sty	15
2.3.6. utopia.sty	15
2.3.7. Obtenir des lettres grecques droites	15
2.3.8. euler.sty & eulervm.sty	16
2.3.9. ae.sty, aecompl.sty et zefonts.sty	16

Table des matières

3. Mélanges	17
3.1. Flottants – « figure », « table »	17
3.2. Annexes	18
3.3. Mathématiques	18
3.4. L'utilisation de <code>\graphicspath</code>	19
3.5. Macros spécifiques à une langue : <code>\...name</code>	21
Annexe : exemple de <code>\sloppy</code>	23
L'utilisation de <code>\graphicspath</code> (version allemande)	23

1. « Péchés mortels »

Dans cette section, nous avons recueilli les erreurs sans doute les plus graves qui apparaissent très régulièrement sur `de.comp.text.tex`, laissant les habitués devant cette alternative : devenir rouges de colère ou fondre en larmes.

1.1. `a4.sty`, `a4wide.sty`

On ne devrait plus utiliser *aucune* de ces deux extensions. Vous devriez les éliminer sans les remplacer dans vos sources \LaTeX . Elles sont remplacées par l'option de classe `a4paper` pour les classes standard, ou par l'option analogue `paper=a4` pour KOMA-Script à compter de la version 3. La mise en page que produisent ces extensions est plus que critiquable d'un point de vue typographique. Pire, il y a plus d'une version de ces extensions dans la nature et toutes ces versions sont incompatibles entre elles. Elles fournissent différentes dispositions pour l'empagement. On ne peut même pas être sûr qu'un document, compilé sur un autre système, aura le même – mauvais – aspect.

1.2. Modifier la mise en page

Les marges produites par les classes standard – `article.cls`, `report.cls`, `book.cls` – semblent souvent trop grandes. Dans ce cas vous devriez utiliser les classes – `scrartcl.cls`, `scrreprt.cls`, `scrbook.cls` – du paquet KOMA-Script, ou bien l'extension `typearea.sty` qui est contenue dans KOMA-Script. La documentation incluse `scrguide` [4] dans le paquet donne de nombreuses informations complémentaires.

Si l'on doit vraiment utiliser un empagement différent de celui produit par l'extension `typearea.sty`, il faut alors plutôt utiliser l'extension `geometry.sty`. Il ne faut pas tenter de modifier la mise en page à coups de `\oddsidemargin & Cie`. L'extension `ansysize.sty` est obsolète, et l'extension `vmargin.sty`, autrefois recommandée, devrait être évitée, car elle peut produire des effets secondaires indésirables⁶.

Il ne faudrait changer `\hoffset` ou `\voffset` sous aucun prétexte, à moins que de comprendre parfaitement le fonctionnement interne de \TeX .

1.3. Changer les extensions et les classes

Il ne faut jamais changer directement les classes \LaTeX – par exemple `article.cls`, `scrbook.cls`, &c. – ou les extensions-fichiers de style, par exemple `varioref.sty`, `color.sty` ! Ou bien on écrit une « classe-conteneur », ou son propre fichier `.sty`, ou bien on *copie* les fichiers de classe ou de style, on modifie la *copie* et on la sauvegarde dans un fichier *différent* avec un nom de fichier *différent*.

Pour un exemple d'écriture d'une classe-conteneur, voyez De-TeX-FAQ sur l'adaptation des classes de documents⁷, ou UK FAQ [3] (« Installing \LaTeX files », section J).

6. Cf aussi Message-ID: <uirrzd16.fsf@uwe-siart.de>

7. <http://projekte.dante.de/DanteFAQ/Dokumentenklasse>

Indication. En général, de telles classes et extensions supplémentaires devraient être installées dans l'arborescence \TeX locale ou bien dans l'arborescence personnelle (présente dans le répertoire $\$HOME$ des systèmes Unix), afin que les changements ne soient pas écrasés lors d'une mise à jour de votre distribution \TeX . Si ces modifications sont utilisées pour un projet spécifique, que vous avez l'intention de transmettre, elles peuvent aussi bien être sauvegardées dans le répertoire de travail du projet.

1.4. Modification de l'interligne avec `\baselinestretch`

De manière générale, les modifications des paramètres devraient s'effectuer au niveau le plus haut possible. Par conséquent, les modifications de l'interlignage peuvent s'effectuer à trois niveaux :

1. Utiliser l'extension `setspace.sty`.
2. Utiliser la commande \LaTeX `\linespread{<Facteur>}`.
3. Redéfinir `\baselinestretch`.

Une redéfinition de paramètres comme `\baselinestretch` opère au plus bas niveau, et devrait donc être réservée aux extensions. Une meilleure méthode, précisément parce qu'elle a été expressément prévue pour cela, consiste à utiliser la commande `\linespread{<Facteur>}`. La meilleure façon de procéder est cependant d'utiliser l'extension `setspace.sty`, qui veille, lors de modifications de l'interlignage, à maintenir les espacements corrects dans les notes de bas de page ou les environnements de listes, pour lesquels les modifications ne sont le plus souvent pas souhaités.

Si donc, on a besoin d'un interligne plus important (par exemple, 1,5 ou 2), on recourra avant tout à l'extension `setspace.sty`. Si en revanche, on ne souhaite faire que des ajustements minimes pour une police de caractères autre que Computer Modern (par exemple Palatino), alors il sera plus judicieux d'employer `\linespread{<Facteur>}`.

Dans le cas de Palatino, on pourra utiliser par exemple `\linespread{1.05}`.

1.5. Indentation et espace entre les paragraphes – `\parindent`, `\parskip`

Il peut être parfois judicieux de changer le renfoncement – ou alinéa⁸ – c'est-à-dire le retrait de la première ligne d'un paragraphe – `\parindent`. On devrait cependant respecter ce qui suit :

- la modification de l'alinéa s'exprime avec une unité dépendant du caractère (« em ») et non avec une unité absolue (« mm »). Cette « dépendance » signifie en l'occurrence que, non seulement l'alinéa varie avec la taille de la police, mais aussi avec la police employée, puisque la largeur de 1 em est adaptée au cas par cas, pour chaque police, par son créateur.

8. Appelé encore, improprement, par décalque de l'anglais : « indentation »[T_{ds}].

- il faut utiliser la syntaxe de \LaTeX , car c'est elle qui pose le moins de problèmes. Par exemple, elle est plus facile à « parser⁹ » pour un programme ou un script externe, la maintenance en est plus simple pour l'utilisateur, et il n'y a pas d'incompatibilité avec d'autres extensions comme `calc.sty`.

▷ **Remplacez :**

```
\parindent=1em
```

par :

```
\setlength{\parindent}{1em}
```

Toutefois si à l'alinéa vous préférez un espace vertical additionnel pour marquer le début d'un nouveau paragraphe, *n'utilisez pas* simplement :

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{\baselineskip}
```

car `\parskip` a aussi une incidence sur les environnements de liste, la table des matières et les en-tête.

L'extension `parskip.sty` et les classes KOMA-Script s'efforcent néanmoins d'éviter ces effets secondaires. Pour savoir comment utiliser les options de KOMA-Script (`parskip`, `halfparskip`, &c.), voir [scrguien \[4\]](#). Lorsque l'on utilise une des classes de KOMA-Script on *ne doit pas* charger `parskip.sty` par surcroît.

1.6. Mettre en évidence les formules mathématiques avec `$$... $$`

S'il vous plaît, ne le faites pas ! `$$... $$` est une commande `plainTeX` qui devrait être évitée en \LaTeX . Cela rendrait incohérents les espacements verticaux des formules ainsi mises en évidence, — voir aussi la section 3.3 p. 18 ; notez l'avertissement concernant `displaymath` en rapport avec l'extension `amsmath.sty`. De plus, l'option de classe `fleqn` ne fonctionne plus dans un tel cas.

▷ **Remplacez :**

```
$$ ... $$
```

par :

```
\[ ... \]
ou (seulement avec amsmath.sty)
\begin{equation*}
...
\end{equation*}
ou bien (seulement sans amsmath.sty)
\begin{displaymath}
...
\end{displaymath}
```

9. Faire l'analyse syntaxique, décomposer.

1.7. `\def` & `\newcommand`

Les macros devraient *toujours* être définies à l'aide de `\newcommand{<name>}{...}` et *pas* avec `\def<name>{...}`¹⁰.

Le problème principal avec `\def` est qu'aucun contrôle de l'existence d'une macro de même nom n'est effectué, de sorte qu'une macro définie précédemment peut être remplacée sans le moindre avertissement.

Des macros déjà existantes peuvent être redéfinies à l'aide de `\renewcommand{<name>}{...}`.

Celui qui sait précisément *pourquoi* il a besoin d'utiliser `\def`, celui-là en connaîtra aussi les avantages et les inconvénients, et peut ne pas tenir compte de cette section.

1.8. Utilisation de `\sloppy`

De manière générale, le commutateur `\sloppy` ne devrait jamais être utilisé, en particulier dans le préambule du document. Pour résoudre des problèmes de coupure de ligne dans certains paragraphes, il faut :

1° vérifier si les motifs de césure corrects (par exemple à l'aide de `babel.sty`), et les polices T1 ont été chargés – pour l'allemand (n)german.sty (voir aussi la FAQ sur la coupure des mots¹¹ ou UK FAQ [3], « Hyphenation », section P.7) et pour le français french de `babel.sty` ;

2° utiliser l'extension `microtype.sty`. Celle-ci modifie légèrement l'espacement entre caractères afin d'améliorer la mise en page. Cette modification automatique n'est possible qu'avec pdfTeX à partir de la version 1.20 et ne fonctionne que pour la création d'un fichier PDF, mais pas si pdfTeX fonctionne en mode DVI ;

3° reformuler le texte. Il n'est pas toujours nécessaire de changer la phrase dans laquelle apparaît le problème de coupure de ligne. Souvent il suffit de changer/déplacer la phrase précédente ou la suivante ;

4° ajuster modérément les paramètres que TeX utilise pour les calculs des coupures de ligne. Axel Reichert a posté¹² une fois, sur de.comp.text.tex, ses ajustements personnels, avec lesquels il évite les principaux problèmes de coupure de ligne tout en préservant une bonne mise en page. (On doit à partir de là faire vraiment attention aux avertissements qui apparaissent et les supprimer en reformulant le texte.)

10. Voir aussi [5], section 2.7.2, et [7], section 3.4.

11. <http://projekte.dante.de/DanteFAQ/Silbentrennung>

12. Voir le message Message-ID: <a84us0\$plqcm\$7@ID-30533.news.dfncis.de>.

2. Commandes, classes et extensions obsolètes

Markus Kohm a écrit un *script* Perl avec lequel on peut chercher en ligne (<http://kohm.de.tf/markus/texidate.html>) les erreurs les plus courantes dans ses fichiers.

Il est cependant à noter que ce *script* n'est pas un parseur complet de \TeX et par conséquent ne cherchera que les erreurs les plus évidentes. Veuillez vérifier votre fichier avant de demander secours dans un forum ou sur une liste de diffusion. D'abord tester, poster ensuite.

2.1. Commandes

2.1.1. Modification du style de caractères

La table 1 présente en vis-à-vis les commandes obsolètes et les commandes actuelles de \LaTeX pour changer le style des caractères. Les macros appelées « locales » ne s'appliquent qu'à leur argument, tandis que celles appelées « globales » ou « commutateurs » s'appliquent à tout le texte jusqu'à la fin du groupe dont elles font partie.

TABLE 1 – Commandes pour changer le style de caractères

Obsolète	Remplacé en $\text{\LaTeX}_{2\epsilon}$ par	
	local	global/commutateur
<code>{\rm ...}</code>	<code>\textrm{...}</code>	<code>\rmfamily</code>
<code>{\sf ...}</code>	<code>\textsf{...}</code>	<code>\sffamily</code>
<code>{\tt ...}</code>	<code>\texttt{...}</code>	<code>\ttfamily</code>
<code>{\it ...}</code>	<code>\textit{...}</code>	<code>\itshape</code>
<code>{\sc ...}</code>	<code>\textsc{...}</code>	<code>\scshape</code>
<code>{\sl ...}</code>	<code>\textsl{...}</code>	<code>\slshape</code>
<code>{\bf ...}</code>	<code>\textbf{...}</code>	<code>\bfseries</code>
—	<code>\textmd{...}</code>	<code>\mdseries</code>
—	<code>\textup{...}</code>	<code>\upshape</code>
—	<code>\emph{...}</code>	<code>\em^a</code>

^a Utile dans les définitions de macros. Dans un texte, ce n'est pas `{\em ...}` qu'on doit employer, mais `\emph{...}`.

Pourquoi ne faut-il pas utiliser de commandes obsolètes? Les commandes obsolètes ne supportent pas le système de gestion des fontes¹⁴ de $\text{\LaTeX}_{2\epsilon}$. Par exemple, `{\bf foo}`, annule tous les attributs de fonte déjà spécifiés avant d'imprimer *foo* en gras. C'est pourquoi vous ne pouvez pas obtenir de l'italique gras à l'aide de

14. NFSS : New Font Selection Scheme, ou « Nouveau Schéma de Sélection des Fontes ». Le fntguide [6] en donne une vue d'ensemble.

`{\it \bf Test}` (la définition mentionnée produit **Test**). En revanche, les nouvelles commandes `\textbf{\textit{Test}}` se comporteront comme escompté et produiront : *Test*. En outre, avec les anciennes commandes, il n’y a pas de « correction italique » : comparez par exemple *neufhampes* (`{\it neuf}hampe`) et *neufhampes* (`\textit{neuf}hampes`).

2.1.2. Fractions mathématiques (`\over` & `\frac`)

On devrait éviter la commande `\over`. `\over` est une commande T_EX qui est difficile à analyser ou qui ne peut pas être analysée du tout du fait de sa syntaxe très différente de celle de L^AT_EX. En particulier, l’extension `amsmath.sty` redéfinit `\frac{}{}` et l’utilisation de `\over` produira des messages d’erreur. Un autre argument en faveur de `\frac{}{}` est qu’il est plus facile de composer le numérateur et le dénominateur d’une fraction, notamment pour des fractions un peu complexes.

▷ **Remplacez :**

```
$a \over b$
```

par :

```
$\frac{a}{b}$
```

2.1.3. Centrer du texte avec `\centerline`

La commande `\centerline` est une autre commande T_EX que l’on devrait éviter en L^AT_EX. D’une part `\centerline` est incompatible avec quelques extensions de L^AT_EX (par exemple `color.sty`); d’autre part, elle peut produire des effets indésirables ou inattendus :

```
\begin{enumerate}
\item \centerline{Premier point}
\end{enumerate}
1. Premier point
```

▷ **Remplacez :**

```
\centerline{...}
```

par :

```
{\centering ...}
ou bien
\begin{center}
...
\end{center}
```

Note. Pour des précisions sur le centrage de graphiques et de tables, voir section 3.1 p. 17.

2.2. Fichiers de classes et d'extensions

2.2.1. `scrletter.cls` & `scrlttr2.cls`

La classe `scrletter.cls` du paquet KOMA-Script est obsolète et a été remplacée par `scrlttr2.cls`. Afin de produire une mise en page *semblable* à celle de l'ancienne classe de lettre de KOMA-Script on peut utiliser l'option de classe KOMAold qui fournit un mode de compatibilité.

▷ **Remplacez :**

```
\documentclass{scrletter}
```

par :

```
\documentclass[KOMAold]{scrlttr2}
```

Note. Pour de nouveaux modèles de lettres, on devrait utiliser la nouvelle interface, qui est bien plus flexible.

Développer les différences entre les deux interfaces utilisateurs sort du cadre de cet aperçu. Nous renvoyons à [scrguien \[4\]](#) pour les détails.

2.2.2. `epsf.sty`, `psfig.sty`, `epsfig.sty` & `graphics.sty`, `graphicx.sty`

Les extensions `epsf.sty` et `psfig.sty` ont été remplacées par les extensions `graphics.sty` et `graphicx.sty` respectivement. `epsfig.sty` est juste une enveloppe¹⁵ pour permettre d'utiliser l'extension `graphicx.sty` avec d'anciens documents écrits avec `psfig.sty`.

Comme `epsfig.sty` utilise `graphicx.sty` en interne, on *peut* encore utiliser `epsfig.sty`. On ne devrait pas, toutefois, l'utiliser pour de nouveaux documents. Pour cela les extensions `graphics.sty` ou `graphicx.sty` sont préférables. `epsfig.sty` est fourni essentiellement pour assurer la compatibilité comme mentionné ci-dessus.

Pour les différences entre `graphics.sty` et `graphicx.sty`, voyez [grfguide \[2\]](#). Pour des précisions sur le centrage des graphiques voyez section 3.1 p. 17.

15. Une enveloppe, anglais *wrapper*, désigne ici un fichier de style qui lui-même charge un ou plusieurs fichiers de style et, ce faisant, donne accès à leurs fonctionnalités.

▷ **Remplacez :**

```
\usepackage{psfig}  
\psfig{file=image,...}
```

par :

```
\usepackage{graphicx}  
\includegraphics[...]{image}
```

2.2.3. **doublespace.sty & setspace.sty**

Pour changer l'interlignage, utilisez l'extension `setspace.sty`. L'extension `doublespace.sty` est obsolète, et remplacée précisément par `setspace.sty`. Voyez aussi la section 1.4 p. 2.

▷ **Remplacez :**

```
\usepackage{doublespace}
```

par :

```
\usepackage{setspace}
```

2.2.4. **fancyheadings.sty, scrpage.sty & fancyhdr.sty, scrpage2.sty**

L'extension `fancyheadings.sty` a été remplacée par `fancyhdr.sty`. Une autre manière de modifier les entêtes est fournie par l'extension `scrpage2.sty` du paquet KOMA-Script. Notez, une fois encore, que `scrpage.sty` ne doit pas être utilisé. Pour une documentation sur `scrpage2.sty`, voir [scruien \[4\]](#).

▷ **Remplacez :**

```
\usepackage{fancyheadings}
```

par :

```
\usepackage{fancyhdr}
```

▷ **Remplacez :**

```
\usepackage{scrpage}
```

par :

```
\usepackage{scrpage2}
```

2.2.5. La famille d'extensions `caption.sty`

L'extension `caption2.sty` ne doit plus être utilisée, car il existe une nouvelle version (V.3x) de l'extension `caption.sty`. Prendre soin toutefois de ne pas utiliser une version trop ancienne de l'extension. Pour s'en assurer, charger l'extension comme suit :

▷ **Remplacez :**

```
\usepackage{caption}
```

par :

```
\usepackage{caption}[2008/08/24]
```

Si l'on a utilisé auparavant `caption2.sty`, il faut absolument prendre connaissance de la nouvelle documentation [[12](#), Section C.2].

2.2.6. isolatin.sty, umlaut.sty & inputenc.sty

Quelques notes générales. Il y a essentiellement quatre façons de saisir les *umlauts* allemands et les autres caractères non-ASCII¹⁶.

1. $H\{\backslash"u\}l\le$ ($\{\backslash'E\}$ tage). L'avantage de cette saisie est qu'elle fonctionne toujours et sur n'importe quel système.

Ses inconvénients, toutefois, sont que le crénage¹⁷ entre les lettres est gravement perturbé ; qu'elle est très pénible à saisir à tout le moins dans un texte en allemand ; enfin que le code obtenu est fort peu lisible.

Aussi cette variante devrait *toujours* être évitée, du fait des problèmes créés pour le crénage.

2. Avec $H\backslash"u\}l\le$ ou $H\backslash\{u\}l\le$ ($\backslash'E$ tage ou $\backslash\{E\}$ tage) les problèmes mentionnés ci-dessus (crénage) n'apparaissent pas. Cette saisie est elle aussi utilisable sur tous les systèmes.

L'inconvénient est que la saisie est aussi laborieuse, et que le code obtenu est encore moins lisible que la variante précédente.

Cette variante est la plus judicieuse dans des définitions de macros ou dans des fichiers de style car elle est indépendante du codage et de toute extension.

3. Avec l'extension (n)german.sty ou l'option (n)german de l'extension babel.sty, on peut saisir les *umlauts* allemands en tapant simplement $H"u\}l\le$. Une fois de plus, l'avantage est que cela fonctionne avec tous les systèmes. Comme les extensions babel.sty ou (n)german.sty sont disponibles sur tous les systèmes \TeX , il ne devrait pas non plus y avoir de problème de compatibilité.¹⁸

L'inconvénient est, encore, que la saisie est malcommode et le code-source est plutôt difficile à lire.

Cette variante s'utilise relativement bien pour le corps du texte. Mais on devrait l'éviter dans les définitions de macro ou les préambules.

4. L'entrée directe – $H\ddot{u}l\le$ (Étage). Les avantages de cette variante sont patents. Le texte-source s'écrit et se lit « normalement ».

L'inconvénient est qu'il faut dire à \LaTeX quel est le codage d'entrée utilisé et qu'il peut aussi y avoir des problèmes lors de l'échange de fichiers entre systèmes différents. Cela *n'est pas* un problème pour \TeX ou \LaTeX eux-mêmes, mais cela peut créer des *difficultés d'affichage du texte-source dans les éditeurs de texte* sur des systèmes divers. Par exemple, le symbole de l'euro codé en iso-8859-15 (latin9) peut être *représenté* par € dans un éditeur sous Windows (CP1252).

Cette variante fonctionne très bien pour le corps du texte. Il vaut mieux, cependant, l'éviter dans les définitions de macros ou les préambules.

16. Comme les lettres accentuées françaises [T_{ds}].

17. Espace positive ou négative insérée entre deux caractères en fonction du couple de caractères.

18. Il n'y a pas d'équivalent à cette saisie pour les accents du français avec babel.sty et french [T_{ds}].

En résumé, dans les macros, préambules et fichiers de style, on devrait utiliser $\text{H}\backslash\text{"ulle}$ ou $\text{H}\backslash\{\text{u}\}\text{lle}$, tandis qu'ailleurs dans le texte on peut utiliser soit $\text{H}\text{"ulle}$ soit $\text{H}\text{ülle}$.

Codage d'entrée. Pour dire à $\text{L}\text{A}\text{T}\text{E}\text{X}$ quel codage d'entrée est employé dans un fichier, il *ne faut pas* utiliser les extensions `isolatin1.sty`, `isolatin.sty` ou `umlaut.sty` ! Ces extensions sont obsolètes ou ne sont pas disponibles sur tous les systèmes.

On doit utiliser l'extension `inputenc.sty` avec les options suivantes¹⁹ :

latin1/latin9 pour les systèmes de type unix (`latin1` fonctionne aussi sous Microsoft Windows et Mac OS X);

ansinew pour Microsoft Windows;

applemac pour Mac²⁰;

cp850 pour OS/2;

utf8 pour les systèmes de type Unix « modernes » et plus généralement pour les systèmes d'exploitation (et les éditeurs) qui supportent Unicode (UTF-8)²¹.

▷ **Remplacez :**

```
\usepackage{isolatin1}
```

par :

```
\usepackage[latin1]{inputenc}
```

▷ **Remplacez :**

```
\usepackage{umlaut}
```

par :

```
\usepackage[latin1]{inputenc}
```

Note. La méthode indiquée ci-dessus pour déclarer le codage d'entrée s'emploie avec les moteurs $\text{L}\text{A}\text{T}\text{E}\text{X}$ « classiques », comme `pdf $\text{L}\text{A}\text{T}\text{E}\text{X}$` . Les développements récents que constituent `X $\text{L}\text{A}\text{T}\text{E}\text{X}$` et `Lua $\text{L}\text{A}\text{T}\text{E}\text{X}$` traitent les fichiers unicode sans qu'on ait à déclarer l'encodage. Tous deux peuvent déjà être utilisés en production, mais ils requièrent en règle générale qu'on récrive les « vieux » fichiers $\text{L}\text{A}\text{T}\text{E}\text{X}$ pour la partie concernant le choix de la langue et des polices de caractère, et ils n'atteignent pas encore les mêmes possibilités que `pdf $\text{L}\text{A}\text{T}\text{E}\text{X}$` en matière de microtypographie.

2.2.7. `t1enc.sty` & `fontenc.sty`

L'extension `t1enc.sty` est obsolète et devrait par conséquent être remplacée par `fontenc.sty`.

19. pour ce qui nous occupe ici [T_{ds}].

20. Sous OS X, `latin1/latin9` devrait être utilisé, car ce codage est plus approprié pour des échanges de fichiers avec des utilisateurs d'autres systèmes d'exploitation. Cependant, il faut impérativement vérifier l'encodage utilisé par son éditeur de texte.

21. C'est le cas de Windows depuis Windows XP et de MacOS X. (B. A.)

▷ **Remplacez :**

```
\usepackage{t1enc}
```

par :

```
\usepackage[T1]{fontenc}
```

2.2.8. natdin.bst & dinat.bst

L'auteur de dinat.bst recommande de remplacer dinat.bst par natdin.bst car le développement de dinat.bst a été arrêté et natdin.bst a été grandement amélioré.

▷ **Remplacez :**

```
dinat.bst
```

par :

```
natdin.bst
```

Note. Dans les versions précédentes de ce document, nous avons fait la recommandation exactement inverse ; celle-ci se fondait sur l'état du développement de l'époque. Les améliorations apportées depuis à natdin.bst justifient maintenant la préférence accordée à cette dernière extension.

2.2.9. glossary.sty & glossaries.sty

L'extension glossary.sty est obsolète ; son auteur lui a fourni pour successeur l'extension glossaries.sty. Des détails sur les changements se trouvent dans [13].

2.2.10. SIstyle.sty, SIunits.sty & siunitx.sty

Les extensions SIstyle.sty et SIunits.sty, utiles à la composition des unités physiques, ne sont plus développées. Les fonctionnalités des deux extensions ont été réunies dans leur successeur siunitx.sty. Sa grande souplesse permet à siunitx.sty de remplacer aussi des extensions comme unitsdef.sty, units.sty, fancyunits.sty et fancynum.sty.

▷ **Remplacez :**

```
\usepackage{SIunits}  
\unit{1}{\metre}
```

par :

```
\usepackage{siunitx}  
\SI{1}{\metre}
```

2.2.11. subfigure.sty, subfig.sty & subcaption.sty

Les extensions subfigure.sty et subfig.sty ne sont plus développées et sont incompatibles avec quelques extensions actuelles comme hyperref.sty et caption.sty. En remplacement, il est conseillé d'utiliser subcaption.sty.

▷ **Remplacez :**

```
\usepackage{subfig}
\begin{figure}
\subfloat[][Text]{Bild}
\end{figure}
```

par :

```
\usepackage{caption}
\usepackage{subcaption}
\begin{figure}
\begin{subfigure}{Breite}
Bild\caption{Text}
\end{subfigure}
\end{figure}
```

2.3. Caractères

Le sujet « fontes et \LaTeX » est, sur de.comp.text.tex, une source intarissable d'« allégresse »²², la plupart du temps suscitée par la question : « Pourquoi les caractères apparaissent-ils si pixellisés dans Adobe Reader ? ». Les réponses *fausses* les plus fréquentes renvoient aux extensions `times.sty` ou `pslatex.sty`. Toutefois, avec ces extensions, non seulement on utilise des polices complètement différentes, mais, comme ces extensions sont obsolètes, on soulève de nouveaux problèmes. Si, comme recommandé dans la section 2.3.9, on utilise²³ `Latin Modern` pour remplacer (quasi) à l'identique la police « standard » de \LaTeX (c'est-à-dire `Computer Modern`), alors l'aspect pixellisé des caractères dans Adobe Reader disparaît.

Le `fntguide` [6] donne des informations générales sur le « New Font Selection Scheme » (NFSS) de $\LaTeX 2_{\epsilon}$

2.3.1. times.sty

L'extension `times.sty` est obsolète (voir `psnfss2e` [10]). Elle définit `\rmdefault` comme étant la police « Times », `\sfdefault` comme « Helvetica » et `\ttdefault` comme « Courier », sans toutefois leur associer les polices mathématiques correspondantes. De plus, Helvetica n'est pas réduite correctement et apparaît trop grande. Si l'on désire utiliser la combinaison Times/Helvetica/Courier il faut alors faire ce qui suit :

▷ **Remplacez :**

```
\usepackage{times}
```

par :

```
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
```

Note. Le facteur de réduction de `helvet.sty`, combinée avec Times, devrait être pris entre 0,90 et 0,92.

22. Ainsi que sur fr.comp.text.tex et comp.text.tex d'ailleurs [T_{ds}].

23. Via l'extension `lmodern.sty` [T_{ds}].

2.3.2. mathptm.sty

L'extension mathptm.sty est le prédécesseur de mathptmx.sty.

▷ **Remplacez :**

```
\usepackage{mathptm}
```

par :

```
\usepackage{mathptmx}
```

2.3.3. pslatex.sty

L'extension pslatex.sty fonctionne en interne comme mathptm.sty + helvet.sty (réduit). Toutefois, elle utilise un Courier trop étroit. Le principal inconvénient de pslatex.sty est qu'il *ne fonctionne pas* avec les codages T1 et TS1.

▷ **Remplacez :**

```
\usepackage{pslatex}
```

par :

```
\usepackage{mathptmx}  
\usepackage[scaled=.92]{helvet}  
\usepackage{courier}
```

Note sur les combinaisons de Times & Helvetica. On peut aussi bien en rester à la police cmtt pour une police type « machine à écrire », et ne pas charger du tout l'extension courier.sty.

2.3.4. palatino.sty

L'extension palatino.sty se comporte comme times.sty – hormis qu'elle définit naturellement `\rmdefault` comme « Palatino ». C'est pourquoi on ne devrait plus l'utiliser.

▷ **Remplacez :**

```
\usepackage{palatino}
```

par :

```
\usepackage{mathpazo}  
\usepackage[scaled=.95]{helvet}  
\usepackage{courier}
```

Note. Le facteur de réduction pour helvet.sty en combinaison avec la fonte Palatino devrait être fixé à 0,95.

« Helvetica » *n'est pas* la meilleure linéale à associer à « Palatino » et doit être vue, autant que possible, comme une « vieille » association, même si elle a longtemps constitué le complément *disponible librement* le mieux assorti à « Palatino ». Depuis lors, on dispose des polices Vera/Bera Sans – à notre avis un meilleur choix.

Celui qui possède un CD CorelDraw[®], même ancien²⁴, peut aussi très bien utiliser « Palatino » avec « Frutiger²⁵ » ou « Optima²⁶ ». Walter Schmidt a publié sur sa page personnelle²⁷ les outils nécessaires pour utiliser ces polices (extensions bfr.sty et bop.sty).

2.3.5. mathpple.sty

Cette extension est le prédécesseur de mathpazo.sty. Il lui manque quelques symboles et ils sont pris dans la fonte Euler. D'autres symboles ne conviennent pas bien à Palatino et quelques métriques de la fonte sont incorrectes. Pour plus de précisions, voir psnfss2e [10].

▷ **Remplacez :**

```
\usepackage{mathpple}
```

par :

```
\usepackage{mathpazo}
```

2.3.6. utopia.sty

L'extension utopia.sty ne devrait plus être utilisée, car son successeur fourier.sty offre une meilleure interface avec la police Utopia. Pour plus de précisions, voir psnfss2e [10].

▷ **Remplacez :**

```
\usepackage{utopia}
```

par :

```
\usepackage{fourier}
```

2.3.7. Obtenir des lettres grecques droites

Les passages marqués en rouge ci-après ne sont pas obsolètes dans le sens où « vous ne devriez plus utiliser cela », mais de nos jours, l'édition d'un texte est simplifiée par l'extension *upgreek.sty*^{28 29}. Pour des précisions sur son utilisation, consultez comme toujours la documentation upgreek [9].

24. Les polices fournies avec les versions anciennes de CorelDraw[®] ne contiennent pas encore de symbole euro !

25. Bitstream « Humanist 777 ».

26. Bitstream « Zapf Humanist ».

27. Fonts für T_EX : <http://cq131a.de/>

28. L'extension fourier.sty admet une option upright qui donne les lettres grecques bas de casse, les capitales grecques et romaines droites en mode mathématique [T_{ds}].

29. Les extensions kpfonts.sty, mathdesign.sty et MinionPro.sty ont les mêmes possibilités que fourier.sty (B. A.)

Les trucs de pifont.sty

▷ **Remplacez :**

```
\usepackage{pifont}
\newcommand{\uppi}{%
  \Pisymbol{psy}{112}}
...
\uppi

ou bien

\newcommand[1]{\upgreek}{%
  \usefont{U}{psy}{m}{n}#1}
...
\upgreek{p}
```

par :

```
\usepackage{upgreek}
...
$\uppi$
```

Les trucs de babel.sty

▷ **Remplacez :**

```
\usepackage[greek,...]{babel}
\newcommand{\upgreek}[1]{%
  \foreignlanguage{greek}{#1}}
...
\upgreek{p}
```

par :

```
\usepackage{upgreek}
...
$\uppi$
```

2.3.8. euler.sty & eulervm.sty

L'extension `euler.sty` devrait être remplacée par `eulervm.sty`, car celle-ci règle des problèmes de compatibilité avec d'autres extensions et apporte quelques améliorations de détail :

- `\hbar` (`\hslash` dans cette police) fonctionne maintenant correctement ;
- des caractères mathématiques gras, y compris des symboles grecs sont disponibles.

Pour des informations plus précises, voir `eulervm` [8].

▷ **Remplacez :**

```
\usepackage{euler}
```

par :

```
\usepackage{eulervm}
```

2.3.9. ae.sty, aecompl.sty et zefonts.sty

L'extension `ae.sty` (ainsi que `aecompl.sty`) et `zefonts.sty` servent de solution de fortune pour un problème en relation avec les umlaut et autres lettres accentuées lorsqu'on

3. Mélanges

utilise la police standard de L^AT_EX (Computer Modern). Or celles-ci utilisent le codage OT₁, et donc les lettres accentuées doivent être constituées de deux signes (« ü » est, sommairement, la combinaison de « u » et de « ¨ »). L'algorithme de coupure des mots est perturbé par ces lettres, ce qui entraîne que, par exemple, les mots avec un umlaut ne peuvent être coupés. Les extensions `ae.sty` et `zefonts.sty` remédient à ce problème en recourant à ce qu'on appelle des fontes virtuelles.

Un autre problème, connexe au précédent, n'est cependant pas résolu par les extensions mentionnées : si l'on produit un fichier PDF, les lettres accentuées n'en demeurent pas moins composées de deux signes, ce qui entraîne que la recherche ou le copier-coller dans le document de mots avec un umlaut ne sera pas traité correctement.

Les problèmes que nous venons de décrire ne surviennent plus si l'on utilise une police au codage T₁ et au format « type 1 »³⁰. C'est pourquoi ont été récemment dessinées les polices de caractères `cm-super` et `Latin Modern`. Comme cette dernière est plus conforme à l'originale Computer Modern, il est recommandé de remplacer `CM` par `Latin Modern`.

▷ **Remplacez :**

```
\usepackage{ae}
```

ou

```
\usepackage{zefonts}
```

par :

```
\usepackage[T1]{fontenc}
```

```
\usepackage{lmodern}
```

3. Mélanges

Cette section – à l'exception de [3.2 page suivante](#) – présente des trucs et des précisions plus généraux que ceux regroupés sous le titre de « péchés ».

3.1. Flottants – « figure », « table »

Pour centrer un flottant on devrait utiliser la commande `\centering` plutôt que l'environnement `\begin{center}... \end{center}` car celui-ci introduit un espace vertical additionnel qui n'est la plupart du temps pas souhaité.

30. Bien noter que même si les dénominations T₁ et type 1 se ressemblent, elles ne désignent pas la même chose : T₁ décrit l'encodage des caractères du côté de L^AT_EX, tandis que « type 1 » déclare comment les caractères sont représentés dans le fichier PDF. C'est pourquoi l'utilisation d'une police au codage T₁ résout le problème de coupure des mots avec umlaut, mais pas nécessairement le problème de la recherche des umlauts dans un fichier PDF.

▷ **Remplacez :**

```
\begin{figure}
  \begin{center}
    \includegraphics{image}
  \end{center}
\end{figure}
```

par :

```
\begin{figure}
  \centering
  \includegraphics{image}
\end{figure}
```

Note. Toutefois, lorsqu'il s'agit de centrer une région dans le corps du texte ou dans un environnement `titlepage`, cet espace additionnel peut naturellement être tout à fait souhaité.

3.2. Annexes

Les annexes sont introduites par le commutateur `\appendix`. Ce *n'est pas* un environnement.

▷ **Remplacez :**

```
\begin{appendix}
\section{Blub}
\end{appendix}
```

par :

```
\appendix
\section{Blub}
```

3.3. Mathématiques

De manière générale, pour la typographie mathématique avancée, on devrait utiliser l'extension `amsmath.sty`, éventuellement complétée par `mathtools.sty`³¹. Elle procure de nouveaux environnements qui doivent avant tout remplacer `eqnarray`. Les avantages de l'extension sont les suivants :

- les espacements à l'intérieur et autour des environnements sont plus cohérents ;
- les numéros des équations sont placés de telle sorte qu'ils ne seront plus surchargés par les expressions mathématiques ;
- de nouveaux environnements (par exemple `split`), permettent de mettre en page facilement de longues équations ;
- il est possible de définir facilement de nouveaux opérateurs, analogues à `\sin`, avec des espacements appropriés.

Avertissement. En utilisant l'extension `amsmath.sty`, on ne devrait *en aucun cas* faire usage des environnements `displaymath`, `eqnarray` ni `eqnarray*`, car ils ne sont pas supportés par `amsmath.sty` et il en résulterait des espacements incohérents (comparer les figures 3 et 4 page suivante).

31. Il suffit alors de charger cette dernière (B. A.)

3. Mélanges

$$\begin{array}{l} a = b \\ b = c \\ a = c \end{array}$$

FIG. 3 – Exemple avec `eqnarray*`

$$\begin{array}{l} a = b \\ b = c \\ a = c \end{array}$$

FIG. 4 – Exemple avec `align*`

L’extension `amsmath.sty` adapte correctement `\[...]` et peut donc s’utiliser au lieu de `displaymath`. On peut remplacer `eqnarray` et `eqnarray*` en première approximation par `align` ou `align*`. Pour un panorama complet de `amsmath.sty`, consulter `amslatex` [1].

▷ **Remplacez :**

```
\begin{eqnarray}
  a &=& b \\
  b &=& c \\
  a &=& c
\end{eqnarray}
```

par :

```
\begin{align}
  a &= b \\
  b &= c \\
  a &= c
\end{align}
```

3.4. L’utilisation de `\graphicspath` ³²

Il faut préciser que, pour des raisons de sécurité liées avant tout, mais pas exclusivement, à Windows, les versions les plus récentes de \TeX (que ce soit `MiKTeX` ou `TeX Live` depuis la version 2009) acceptent une version de \TeX réduite, protégée de l’exécution de la commande `\write18` (qui équivaut à donner une instruction : de façon précise, l’argument de `\write18` est interprété comme si c’était une ligne de commande). \TeX permet d’exécuter pleinement un certain nombre d’instructions (qui concernent entre autres `BibTeX`, `epstopdf`, &c.), en exécute d’autres en mode restreint et enfin n’en exécute pas du tout d’autres encore (comme la suppression de fichiers), qui doivent être autorisées par l’utilisateur ³³.

En conséquence, les programmes ne peuvent plus utiliser de chemins absolus (c’est-à-dire qui partent de la racine du disque dur), mais seulement des chemins relatifs au répertoire dans lequel se trouve le fichier principal du document.

La macro fort répandue `\graphicspath` présente l’avantage qu’une fois exécutée, elle

32. Cette section ne correspond pas à la version originale allemande, mais à la traduction italienne, entièrement réécrite par Mauro Sacchetto. La traduction de la version allemande se trouve en annexe.

33. À moins de ne pas éliminer toutes les limitations de `\write18` en lui indiquant explicitement dans la ligne de commande que l’éditeur envoie au système d’exploitation pour lancer des commandes comme `pdflatex : option -shell-escape` de `TeX Live` et `option -enable-write18` de `MiKTeX`. Toutefois, il faut être conscient des risques que comporte une telle opération pour la sécurité du système tout entier.

évite de devoir spécifier à chaque fois le ou les répertoires où se trouvent les images. Elle devrait toutefois être évitée parce qu'elle présente pour le moins des limites :

1. La recherche par \TeX du fichier graphique à traiter dure plus longtemps que si cette recherche était confiée à la librairie `kpathsea`, à laquelle \TeX recourt normalement pour la recherche de fichiers, même si avec les processeurs actuels, l'augmentation de durée est négligeable.
2. Les noms de fichiers sans leur chemin doivent être soigneusement différenciés dans Windows, qui ne distingue pas entre lettres minuscules et majuscules ; cette différenciation doit avoir lieu même si d'éventuels fichiers homonymes se trouvent dans des sous-répertoires.

De tels problèmes ne se posent pas si tous les fichiers graphiques se trouvent dans un seul sous-répertoire de celui qui contient le fichier principal du document ; on peut donc alors continuer d'utiliser `\graphicspath` en toute sécurité. Si en revanche les parties d'un document complexe, par exemple les chapitres, se trouvent chacune dans son propre sous-répertoire, et si chaque partie conserve à son tour ses propres images dans un sous-sous-répertoire, la situation se montre très confuse et inutilement compliquée par rapport à celle que l'on peut avoir en incluant chaque fichier graphique en spécifiant son nom complet avec chemin et extension³⁴.

La raison pour laquelle la recherche d'un fichier graphique dure plus longtemps en utilisant `\graphicspath` résulte du fait que \TeX conserve en interne une liste des chemins spécifiés par cette commande. Dans un document complexe, avec de nombreux chapitres qui requièrent à leur tour différents chemins d'accès à leurs figures, cette recherche doit balayer récursivement tous les chemins d'accès contenus dans la liste en question et regarder dans chaque dossier spécifié par chacun de ces chemins. Si au contraire, on spécifie chaque fichier graphique avec son chemin d'accès, \TeX ne doit plus rien rechercher, mais pointe vers le fichier spécifique, étant donné que le nom complet du fichier est univoque.

Le recours à la variable d'environnement `TEXINPUTS` apparaît également problématique³⁵. Cette variable peut être fixée selon les modalités propres aux divers systèmes d'exploitation. Bien qu'il s'agisse plus d'un ajout que d'une véritable modification de la variable `path` du système, celle-ci est de toute façon permanente et par conséquent sera modifiée pour chaque nouveau document (outre que les anciennes versions de Windows ne comportent pas la possibilité d'avoir séparément une variable `TEXINPUTS` du système et une de l'utilisateur).

34. Avant 2009, il était conseillé de ne pas spécifier l'extension, parce que les formats de fichiers graphiques que `latex` et `pdflatex` pouvaient inclure étaient mutuellement exclusifs, et donc dans le même dossier pouvait figurer une image donnée au format `.eps` aussi bien qu'au format `.pdf`. Aussi, sans spécifier l'extension et sans modifier le fichier-source, `latex` incluait le fichier `.eps`, tandis que `pdflatex` incluait le fichier `.pdf`. Comme, à présent, `pdflatex` peut inclure des fichiers graphiques au format `.eps`, cette façon de procéder n'a plus de raison d'être, et peut même se révéler pernicieuse.

35. Cf. la réponse de David Carlisle au rapport de bogue de Markus Kohm : <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618>.

La méthode la plus sûre et efficace (et valide aussi pour `\graphicspath`) consiste alors à définir des liens symboliques au moyen des commandes classiques `ln -s` pour les systèmes UNIX et `mklink` dans Windows Vista ou Windows 7 (cette méthode est inapplicable avec les versions précédentes de Windows, qui ne connaissent pas les liens symboliques). À la différence de ce qui se produit avec `TEXINPUTS`, de tels liens, étant locaux, n'apportent aucune modification au path du système. Grâce à eux, les répertoires des images paraissent (virtuellement) présents dans le répertoire du document, où qu'ils soient placés physiquement sur le disque dur, et ils sont donc identifiés par `\write18`, partiellement autorisé.

Il est toujours utile de consulter la documentation du système d'exploitation et de la distribution `TeX` utilisés.

3.5. Macros spécifiques à une langue : `\...name`

De temps à autre la question revient sur de.comp.text.tex de comment modifier, par exemple, « Références » en « Liste des ouvrages cités » ou autre entête. Aussi ces macros ont été rassemblées dans le tableau 2 page suivante. Elles sont tirées de l'extension `frenchb.sty`. Plus précisément, elles sont tirées des fichiers `frenchb.lfd` pour `babel.sty` et `gloss-french.lfd` pour `polyglossia.sty`.

Par exemple, pour changer « Table des figures » en « Table des illustrations », on utilisera la commande suivante :

```
\renewcommand*{\listfigurename}{Table des illustrations}
```

Les autres macros se changent de manière analogue. Si l'on utilise `babel.sty`, il faut passer par `\addto`. En effet, chaque changement de langue (dans notre cas le français) appelle la commande : la personnalisation serait donc perdue et le terme employé par défaut réintroduit. La commande `\addto` est définie dans `babel.sty`, mais on peut l'appliquer aussi à `polyglossia.sty`, mais en prenant soin de charger en outre `etoolbox.sty`. Voir aussi [De-TeX-FAQ](#) sur ce thème^{36 37}.

La façon correcte d'écrire est par conséquent :

```
\addto{\captionsfrench}{%
  \renewcommand*{\listfigurename}{Table des illustrations}}
```

Si l'on utilise `KOMA-Script`, il convient d'employer la commande équivalente fournie `\renewcaptionname` :

```
\renewcaptionname{french}{\listfigurename}{Figure}
```

Cette commande doit figurer après `\begin{document}`.

36. « Meine Versuche, `\figurename`, `\tablename` o.ä. bei Verwendung des Pakets `babel.sty` umzudefinieren, mißlingen. Was kann ich tun ? », in <http://projekte.dante.de/DanteFAQ/Verschiedenes>.

37. Et, pour le français, voyez la « Documentation sur le module `frenchb` de Babel » par Daniel Flipo, disponible sur <http://daniel.flipo.free.fr/frenchb/> [T_{ds}].

3. Mélanges

TABLE 2 – Macros définies par *babel.sty* et *polyglossia.sty* avec l'option *french*

Nom de macro	Définition originelle	Sortie habituelle en français
<code>\prefacename</code>	Preface	Préface
<code>\refname^a</code>	References	Références
<code>\abstractname</code>	Abstract	Résumé
<code>\bibname^b</code>	Bibliography	Bibliographie
<code>\chaptername</code>	Chapter	Chapitre
<code>\appendixname</code>	Appendix	Annexe
<code>\contentsname</code>	Contents	Table des matières
<code>\listfigurename</code>	List of Figures	Table des figures
<code>\listtablename</code>	List of Tables	Liste des tableaux
<code>\indexname</code>	Index	Index
<code>\figurename</code>	Figure	Fig.
<code>\tablename</code>	Table	TABLE
<code>\partname</code>	Part	partie
<code>\enclname</code>	encl	P. J.
<code>\ccname</code>	cc	Copie à
<code>\headtoname</code>	To	—
<code>\pagename</code>	Page	p.
<code>\seename</code>	see	<i>voir</i>
<code>\alsoname</code>	see also	<i>voir aussi</i>
<code>\proofname</code>	Proof	Démonstration
<code>\glossaryname</code>	Glossary	Glossaire

^a Seulement dans la classe *article*.

^b Seulement dans les classes *report* et *book*.

Annexe : exemple de `\sloppy`

Voici le code de l'exemple que Markus Kohm a publié :

```
\documentclass{article}

\setlength{\textwidth}{20em}
\setlength{\parindent}{0pt}
\begin{document}
\typeout{First without \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\ -ta\ -tata
tatata tatata tatata tatata tatata tatata tata\ -tata
tatata tatata tatata tatata ta\ -tatatatt\ -ta
tatata tatata tatata tatata tatata tatata ta\ -ta\ -ta\ -ta

\typeout{done.}

\sloppy
\typeout{Second with \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\ -ta\ -tata
tatata tatata tatata tatata tatata tatata tata\ -tata
tatata tatata tatata tatata ta\ -tatatatt\ -ta
tatata tatata tatata tatata tatata tatata ta\ -ta\ -ta\ -ta

\typeout{done.}
\end{document}
```

Source : Message-ID: <8557097.gEimXdBtjU@ID-107054.user.dfncis.de>

L'utilisation de `\graphicspath` (version allemande)

La macro fort répandue `\graphicspath` devrait être évitée et remplacée par la variable d'environnement `TEXINPUTS` pour les raisons suivantes³⁸ :

1. La recherche par \TeX dure davantage que si cette tâche est confiée à la bibliothèque `kpathsea` (avec les processeurs actuels, cela n'a plus la même importance).
2. Chaque image consomme une partie de la mémoire allouée à \TeX et celle-ci n'est plus disponible durant toute la compilation.

Dans un Bourne-shell, on peut utiliser la commande suivante :

³⁸. Cf. la réponse de David Carlisle au rapport de bogue de Markus Kohm : <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618>

```
[luser@host]> TEXINPUTS=Rep_Images: latex datei.tex
```

ou ajouter cette ligne dans `$HOME/.profile` :

```
export TEXINPUTS=./Rep_Images:
```

Cette ligne assure que les fichiers de `Rep_Images` sous le répertoire de travail seront bien trouvés.

Sous Microsoft Windows (≤ 98), on fixe la variable d'environnement au moyen de :

```
set TEXINPUTS=.\Rep_Image;
```

ou bien :

```
set TEXINPUTS=./Rep_Images;
```

pour $\text{MiK}\TeX$ dans le fichier `autoexec.bat`. Dans Windows NT4 et Windows 2000, on peut fixer les variables d'environnement en allant dans Panneau de configuration \rightarrow Système \rightarrow Environnement, et dans Windows XP, Panneau de configuration \rightarrow Système \rightarrow Avancé \rightarrow Variables d'environnement

Ces procédures sont données à titre d'exemples. Nous sommes conscients qu'on peut adapter `TEXINPUTS` de différentes façons ou par le biais d'autres fichiers. On trouvera plus de précisions dans la documentation du système d'exploitation ou de la distribution $\text{T}\TeX$.

Références

- [1] AMERICAN MATHEMATICAL SOCIETY, *User's Guide for the amsmath Package*. Version 2.0, décembre 1999.
URL: [CTAN:macros/latex/required/amslatex/](#)
- [2] David P. CARLISLE, *Packages in the 'graphics' bundle*. Novembre 2005.
URL: [CTAN:macros/latex/required/graphics/](#)
- [3] Robin FAIRBAIRNS, *The UK T_EX FAQ. Your 459 Questions Answered..* Version 3.26, 28 février 2013.
URL: <http://www.tex.ac.uk/faq>
- [4] Markus KOHM et Jens-Uwe MORAWSKI, *KOMA-Script*. Version 3.11, juillet 2012.
URL: [CTAN:macros/latex/supported/koma-script/](#)
- [5] THE L^AT_EX₃ PROJECT TEAM, *L^AT_EX₂ ϵ for class and package writers*. Février 2006.
URL: [CTAN:macros/latex/doc/clsguide.pdf](#)
- [6] THE L^AT_EX₃ PROJECT TEAM, *L^AT_EX₂ ϵ font selection*. Novembre 2005.
URL: [CTAN:macros/latex/doc/fntguide.pdf](#)
- [7] THE L^AT_EX₃ PROJECT TEAM, *L^AT_EX₂ ϵ for authors*. Novembre 2005.
URL: [CTAN:macros/latex/doc/usrguide.pdf](#)
- [8] Walter SCHMIDT, *The Euler Virtual Math Fonts for use with L^AT_EX*. Version 4, janvier 2005.
URL: [CTAN:fonts/eulervm/](#)
- [9] Walter SCHMIDT, *The upgreek package for L^AT_EX₂ ϵ* . Version 2.0, février 2003.
URL: [CTAN:macros/latex/contrib/supported/was/](#)
- [10] Walter SCHMIDT, *Using common PostScript fonts with L^AT_EX*. PSNFSS version 9.2, septembre 2004.
URL: [CTAN:macros/latex/required/psnfss/psnfss2e.pdf](#)
- [11] Marco DANIEL, Patrick GUNDLACH, Walter SCHMIDT, Jörg KNAPPEN, Hubert PARTL et Irene HYNA, *L^AT_EX₂ ϵ -Kurzbeschreibung*. Version 3, juillet 2012.
URL: [CTAN:info/lshort/german/](#)
Dernière version de la traduction française disponible à
URL: [CTAN:info/lshort/french](#)
- [12] Axel SOMMERFELDT, *Customizing captions of floating environments*. Version 3.3, novembre 2011.
URL: [CTAN:macros/latex/contrib/caption/](#)
- [13] Nicola L. C. TALBOT, *Upgrading from the glossary package to the glossaries package*. Novembre 2012.
URL: [CTAN:macros/latex/contrib/glossaries/glossary2glossaries.pdf](#)