

UTILISATION LOCALE DE `\fontspec` POUR DES CARACTÈRES SPÉCIAUX

© Jacques ANDRÉ

RÉSUMÉ. On montre, à partir d'exemples simples, comment `fontspec` permet désormais d'utiliser toutes les fontes du commerce, y compris celles très « simples » comme les fontes de titrage, de dingbats et de façon plus générale de « casseaux » pour imprimer des caractères spéciaux de façon isolée (ou pour des titres).

Ce document a été écrit en 2015 pour une *Lettre GUTenberg* à venir. Aucune *Lettre* n'étant parue depuis, j'ai réécrit cette note et l'ai soumise aux *Cahiers GUTenberg* en 2018 (d'où le « style » de ce document). Mais, fin octobre 2019, comme rien n'est toujours pas paru (ni *Lettre*, ni *Cahier*) et comme je pense que cette note peut rendre quelques services, je la met sur mon site à l'url suivante (et y ferai d'éventuelles corrections locales, avant de la réécrire) :

<http://jacques-andre.fr/fontex/casseau+fontspec.pdf>




Using `\fontspec` to get special characters on a one-off basis

ABSTRACT. Thanks to `fontspec`, any font may be used in L^AT_EX. In the present tutorial, the very single case of « elementary » fonts, such as titling fonts, dingbats, tiles... may be used in a very simple way to print special characters on a one-off basis or in titlings.

TABLE DES MATIÈRES

1. Premiers exemples	4
2. Sélection des fontes par <code>fontspec</code>	7
2.1. La commande brute <code>\fontspec</code>	7
2.2. Appel d'une famille de fontes	8
2.3. Où mettre les fontes casseaux ?	9
3. À propos d'utf-8	10
4. À propos des fontes Unicode	12
4.1. Combinaison de caractères	13
4.2. Caractères des zones privées d'Unicode	14
5. Fontes non-Unicode	15
5.1. Fontes casseau	15
5.2. Ornaments, bordures et pavages	15
6. Fontes de titrage	21
6.1. Modification des caractéristiques des fontes avec <code>fontspec</code>	22
6.1.1. L'option <code>Ligatures=TeX</code>	23
6.1.2. Ligatures communes	24
6.1.3. Crénage	26
7. Changer la fonte des titres	27
8. En guise de conclusion	30
Bibliographie	31
Annexe : inventaire des caractères d'une fonte	31
Remerciements	34

BIEN SOUVENT, il arrive que l'on ait besoin ou envie d'imprimer de façon isolée un caractère qui ne se trouve pas dans la fonte courante utilisée pour le texte d'un document. Ce peut être, par exemple,

- un caractère graphique comme le B de la lettrine du début de ce paragraphe, comme l'appareil photo , le symbole de notation dentaire Φ ou comme le coin de cadre baroque ;
- l'icône  de *Creative Commons Licence*;
- un signe de monnaie comme ₮ (cruzeiro du Brésil) ;
- un caractère ancien comme le signe « ¶ » (paragraphe au xvi^e siècle) ou le symbole alchimique de bismuth ♃ ;
- etc.

On appelle souvent *caractères spéciaux* ces caractères et parfois *fontes-casseaux*¹ les fontes qui les proposent.

En revanche, on ne qualifie pas de « spéciaux » les caractères accentués des langues européennes, ni les idéogrammes des langues asiatiques, ni même les caractères mathématiques, musicaux ou chimiques, etc. Ils font partie des fontes normales et suivent une grammaire spécifique contextuelle (gestion des diacritiques, crénages, ligatures, composition de symboles, mais aussi italiques, gras, petites capitales, etc.). Il s'agit d'ailleurs plutôt de familles de fontes ou, comme on le disait du temps du plomb, de *fontes de labeur*. T_EX a été l'un des tout premiers logiciels à définir un format pour la gestion de ces fontes. Mais sa machinerie était spécifique et obligeait à une certaine gymnastique (pour respecter le format des fontes T_EX, et notamment la création des divers fichiers .pfb, .tfm, map, etc.) ce qui rendait difficile l'utilisation des fontes plus commerciales (utilisant les formats .ps, .ttf, otf, ...). Même pour n'utiliser qu'un seul caractère d'une quelconque fonte de dingbats².

¹Du temps du plomb, les compositeurs mettaient dans une petite casse, appelée *casseau*, les caractères qui n'ont habituellement pas place dans la grande casse. Le terme est aussi resté chez les linotypistes pour les caractères spéciaux qui ne passaient pas dans la machinerie de la Lino mais dont le moule était inséré à la main dans la ligne avant qu'elle ne soit fondue.

²Alors qu'avec Word c'était très facile. Ce n'était pourtant pas très compliqué en T_EX non plus... Pour mes propres besoins, j'avais écrit un petit script Perl

Aussi beaucoup de ces signes « spéciaux » (ou indépendants) ont-ils été rendus accessibles par et pour les utilisateurs de \TeX par le biais de nombreux packages (on en trouvera une liste dans [4]). Mais pas tous ; ou alors pas avec le graphisme voulu. D'où le besoin d'utiliser encore des fontes casseaux. Mais depuis l'arrivée d'une part des « fontes Unicode » (qui, même si elles sont incomplètes, offrent beaucoup de ces caractères « spéciaux ») et, d'autre part, de $\text{Lua}\TeX$ et de $\text{Xe}\LaTeX$ avec leur package `fontspec`, la manipulation des fontes et l'appel à ces caractères spéciaux sont devenus bien plus simples. Toutefois, même si « qui peut le plus peut le moins », la documentation – notamment le manuel `fontspec` [5] – pour être complète traite souvent en détail des choses compliquées au point de faire croire que c'est le cas général. C'est donc le contraire – montrer l'utilisation simple de `fontspec` pour les cas simples – que voudrait aborder cet article « pour les nuls ».

Un autre cas spécial d'utilisation de ces casseaux est celui des grand-titres (par exemple de couverture) mais aussi des titres de sections, sous-sections, etc. Nous en dirons deux mots (en sections 6 et 7).

1. PREMIERS EXEMPLES

La figure 1.1 montre un exemple simple dont la naïveté nécessite quelques commentaires.

1. On n'attachera aucune attention au sens de cet exemple, ni des autres de cet article. Seule compte la façon d'imprimer certains signes !

2. La première ligne est un commentaire reconnu par certains éditeurs de \TeX (dont `TeXShop` sur les Mac) pour ouvrir le fichier source avec le format `utf-8`. Il ne sert plus beaucoup maintenant que le codage par défaut de \LaTeX est `utf-8` (voir section 3). Mais personnellement je continue, en cette longue période de transition de `latin-1` à `Unicode`, de la mettre pour savoir lesquels de mes propres programmes sont déjà en `utf-8`. Et ici, il permet surtout d'indiquer au lecteur de ce *Cahier* que le codage est bien `utf-8`...

pour MacOS permettant de traduire des fontes TTF, OpenType, etc. en `.pfb` et de les utiliser après les avoir mises dans l'arborescence `TDS` ; il est toujours à <http://jacques-andre.fr/fontex/casseaux.pdf>.

```

%!TEX encoding = UTF-8 Unicode
\documentclass{article}
\usepackage{fontspec}
\begin{document}
\Large À Bakou, le kilo de tomates
est à 1,5\,\{\fontspec{Symbola}\char"20BC}
(soit 0,75\,\,€).
\end{document}

```

À Bakou, le kilo de tomates est à 1,5 ₼ (soit 0,75 €).

FIGURE 1.1. Exemple très simple d'utilisation d'un caractère « exotique » grâce à `\fontspec` avec Lua \LaTeX ou Xe \LaTeX .

3. Justement, puisque le codage est utf-8, je n'ai pas eu besoin de coder (en utilisant par exemple la commande `\texteuro`) le symbole € : il m'a suffit de le taper directement depuis mon clavier.

4. L'intérêt de cet exemple est de montrer qu'on a pu imprimer le signe ₼ : c'est le symbole monétaire *manat* en usage dans la République d'Azerbaïdjan. Une recherche rapide montre que le manat n'est, à priori, pas connu dans le monde \LaTeX (du moins dans la liste des symboles monétaires de [4]). En revanche, on trouve sur le web qu'il s'agit du caractère Unicode U+20BC (SYMBOLE MANAT³); certains sites⁴ énumèrent les fontes Unicode proposant ce signe, dont la fonte Symbola.

5. On a donc téléchargé cette fonte Symbola.ttf⁵ dans notre

³J'utilise dans cet article la terminologie française d'Unicode qui est tout ce qu'il y a de plus normative. On trouvera dans [1] la liste des noms français d'Unicode 10.0. Mais cette version, datée d'août 2017, elle, n'est encore qu'à l'état de liste informative (ISO/CEI 14651 a toujours un peu de retard...).

⁴Ici on a utilisé <http://www.fileformat.info/info/unicode/char/20bc/index.htm>.

⁵Librement depuis le site de George Douros, *Unicode Fonts for Ancient Scripts*, qui est à l'url <http://users.teilar.gr/~g1951d/>

répertoire courant (voir section 2.3).

6. Notre programme source ne comprend que trois instructions pour utiliser cette fonte et en imprimer un caractère :

`\usepackage{fontspec}`: dans le prologue pour donner accès à fontspec, le nouveau gestionnaire de fontes ;

`\fontspec{Symbola}`: pour prendre comme fonte courante la fonte Symbola (dans le bloc délimité par la paire `{...}`) ;

`\char"20BC`: instruction traditionnelle de \TeX pour accéder à un caractère de la fonte courante, ici celui de code $20BC_{16}$ (c'est le code Unicode du manat comme on vient de le voir au point 4 ; le signe " indique que ce code est en hexadécimal). On aurait aussi pu utiliser la commande `\symbol{"20BC}`, plus robuste.

Bien sûr on peut utiliser directement le clavier pour taper les caractères. On pourrait ainsi écrire `{\fontspec{Symbola}€}` pour avoir le symbole euro € dans la même fonte que le manat (en Symbola donc) et non € de ce présent texte (ni € de la fonte tt comme dans l'exemple d'appel donné à l'instant).

7. Le reste du texte est composé avec la fonte par défaut Latin Modern Roman, variante OpenType des Computer Modern.

Le point important de cet exemple est qu'on a utilisé un caractère Unicode d'une fonte téléchargée depuis le web et ce sans aucune manipulation ni conversion (par exemple en .pfb).

La figure 1.2 donne un deuxième exemple utilisant la fonte d'une famille fournie avec \TeX Live, DejaVuSans. On n'a pas besoin de la télécharger et fontspec la trouve où elle est (voir section 2.3). On voit en dernière ligne⁶ aussi que la fonte DejaVuSans n'a pas le caractère U+20BC (le manat cité dans le premier exemple) : très peu de fontes offrent la totalité des caractères d'Unicode et c'est un perpétuel soucis de trouver une fonte correspondant à nos propres besoins...

La version utilisée ici est la v.11.00 de juin 2018.

⁶Par le blanc après =... Le caractère de code "20BC n'existant pas dans la fonte DejaVuSans, il est remplacé par fontpsec par le caractère « vide ».

Saisie :	Sortie :
<pre> \documentclass{article} \usepackage{fontspec} \newcommand{\Uex}[1]{% {\fontspec{DejaVuSans} U+#1=\char"#1}} \begin{document} Unicode a désormais des caractères dont le code est supérieur à FFFF, p.ex. \\Uex{1D350}\\ \Uex{1F045}\\Uex{1F435}\\ Mais, pour DejaVu, \Uex{20BC} \end{document} </pre>	<p>Unicode a désormais des caractères dont le code est supérieur à FFFF. p.ex.</p> <p>U+1D350=⋮</p> <p>U+1F045=🗺</p> <p>U+1F435=🌐</p> <p>Mais, pour DejaVu,</p> <p>U+20BC=□</p>

FIGURE 1.2. Appel d'une fonte fournie par T_EXLive.

2. SÉLECTION DES FONTES PAR fontspec

La littérature sur l'usage de fontspec est finalement relativement rare, le manuel de Will Robertson [5] restant la référence⁷.

L'accès aux fontes est géré par le package fontspec qu'il faut appeler dès le prologue par la commande `\usepackage{fontspec}` (après l'éventuel `\usepackage{microtype}`).

2.1. LA COMMANDE BRUTE `\fontspec`

L'instruction qui nous intéresse ici est `\fontspec` même si le manuel [5, p. 9] nous dit que cette commande brute n'est pas recommandée pour les documents car elle est plutôt prévue pour des tests et pour charger des fontes *on a one-off basis* (de façon ponctuelle); mais c'est justement ce que nous voulons faire ici.

La syntaxe générale d'un appel est de la forme

⁷En français on consultera les transparents des exposés de Michel Bovani et de Christophe Caignaert lors de la Journée GUTenberg'2013 : <http://gutenberg.eu.org/IMG/pdf/ChristopheCaignaert.pdf> et <http://gutenberg.eu.org/IMG/pdf/MichelBovani.pdf>.

`\fontspec{nom de la fonte} [choix de propriétés]`

– *nom de la fonte* (en anglais, *font name*) est dans le cas général (voir section 2.2) celui de la famille utilisée. Mais ici, dans le cas d’une fonte brute, il suffit que ce soit le nom du fichier (*file name*) tel qu’on l’a téléchargé (par exemple `Symbola.ttf`), ou celui d’une fonte standard ou présente dans `TeXLive`⁸.

Quand il n’y a pas d’ambiguïté, `fontspec` n’a pas besoin du suffixe précisant le format, et on peut donc écrire `\fontspec{Symbola}` pour utiliser `Symbola.ttf`.

– *choix de propriétés* (en anglais *properties*) sert pour les fontes de labeur (voir section 2.2) mais est inutile ici puisqu’on va s’adresser à des caractères isolés d’une fonte brute qui ne sont pas concernés par des choses contextuelles comme les ligatures, le crénage, etc. ; on pourra toutefois en avoir besoin dans quelques cas particuliers comme les fontes de titrage et on en reparlera alors (section 6.1).

Les deux premières figures (section 1) ont donné des exemples de cette commande, respectivement par nom de fichier (`Symbola`) (que `fontspec` reconnaît être `Symbola.ttf`) et par nom de fonte (`DejaVuSans` pour avoir le fichier par défaut `DejaVuSans-regular.ttf`). Ce sont bien les noms que l’on trouve dans les fichiers `.log`.

2.2. APPEL D’UNE FAMILLE DE FONTES

L’usage « normal » des fontes, c’est-à-dire des fontes de labeur, permet de définir une famille de fontes et de préciser les spécifications choisies et ne relève pas de cet article. En voici donc un seul exemple (figure 2.1) qui montre comment appeler et utiliser la famille TeX Gyre Pagella⁹ comme fonte courante dans ce document (cette fonte est distribuée avec `TeXLive`).

⁸Mais dans ce cas, il faut spécifier explicitement la variante désirée, par exemple `texgyrepagella-bolditalic.otf` ; sauf pour la fonte par défaut de la famille, en général `UprightFont=*-regular` : ainsi `\fontspec{DejaVuSans}` est-il équivalent à `\fontspec{DejaVuSans-regular.ttf}`.

⁹C’est une variante du Palladio d’URW (qui était une copie, légale, du Palatino d’Herman Zapf pour la fonderie Stempel vers 1940), implémentée par GUST (*Grupa Użytkowników Systemu TeX*, notre cousin le groupe polonais de `TeX`) dans

Saisie :

```
\setmainfont{TeX Gyre Pagella}
. . .
Par défaut,
on n'a pas %ici apostrophe droite
besoin de spécifier
ni les \textit{ligatures} fi, fl
ni que \texttt{--} = --,
ni \textbf{etc.}
```

Sortie :

Par défaut, on n'a pas besoin de spécifier ni les *ligatures* fi, fl ni que -- = -, ni etc.

FIGURE 2.1. Exemple d'appel d'une famille de fontes avec fontspec.

En fait `\setmainfont{TeX Gyre Pagella}`¹⁰ revient en gros au même que `\usepackage{tgpagella}` en L^AT_EX classique (c'est-à-dire avec NFSS, *New Font Selection Scheme*, avec *new* depuis 1989...). Pour l'utilisateur, ça revient à peu près au même; mais pour le développeur, il faut définir autant de commandes dans les deux cas (voir, note 7, un pointeur sur l'exemple de la fonte Garamond donné par Michel Bovani). Toutefois, fontspec utilisant les propriétés des fontes .otf, peut plus facilement accéder à certaines propriétés des fontes (notamment aux variantes, aux caractères contextuels comme les fins de mot, etc.). On verra quelques exemples d'utilisation des possibilités (*features*) des fontes en section 6.1.

2.3. OÙ METTRE LES FONTES CASSEAUX ?

Fontspec travaille à peu près comme les fontes de NFSS (à part un mécanisme de cache) : il trouve les fontes à utiliser en cherchant dans l'ordre

- d'abord dans le répertoire courant
- puis dans l'arborescence TeX, avec notamment un accès aux fontes

le cadre du projet T_EX Gyre. Ce même projet propose aussi TeXGyreTermes qui est une version du Times Roman et TeXGyreHeros, un Helvetica; on les utilisera en 7.

¹⁰Avant 2015, il fallait préciser `\setmainfont[Ligatures=TeX]{TeX Gyre Pagella}`; voir section 6.1.1.

de l'année courante de T_EXLive. Par exemple, DejaVuSans est dans `/usr/local/texlive/2018/texmf-dist/fonts/truetype/public/dejavu` (avec T_EXLive 2018).

– Il cherche aussi dans les fontes systèmes (livre des polices du Mac p.ex.).

Mais c'est transparent pour l'utilisateur normal.

Puisqu'on utilise ces fontes casseaux de façon isolée, il est recommandé de les mettre dans le répertoire courant. Mais on peut les regrouper dans un répertoire commun, p.ex. de nom Casseaux. La méthode usuelle est de mettre ce dossier

- soit dans \$TEXMFHOME sous `fonts/truetype/public/Casseaux` ou `fonts/opentype/public/Casseaux`

- soit dans \$TEXMFLOCAL aux mêmes emplacements, mais il faut alors faire régénérer la base de données par un `mktexlsr`.

Sinon, on peut le mettre où on veut, mais il faut alors forcer `\fontspec` à chercher la fonte en suivant le chemin voulu, par exemple en écrivant

- `\fontspec{EnTypo}[Path=/Users/JAndre/MesFontes/Casseaux/]`

3. À PROPOS D'UTF-8

Rappelons d'abord qu'utf-8 est une forme de représentation des codes Unicode utilisant moins d'espace en mémoire que si tous les numéros étaient codés en clair¹¹.

Pour utiliser `fontspec`, les programmes sources doivent être écrits en utf-8 et compilés par LuaL^AT_EX ou X_ƎL^AT_EX. Si l'on part de fichiers écrits initialement, par exemple, en latin-1, il faut d'abord les convertir en utf-8¹².

¹¹Voir le très bon article de Wikipedia à ce sujet : <https://fr.wikipedia.org/wiki/UTF-8>. Notons que si le codage latin-1 est bien un sous-ensemble d'Unicode, ce n'est pas un sous-ensemble d'utf-8, d'où la nécessité de faire un recodage des textes de latin-1 vers utf-8. Le texte ainsi recodé est plus long, mais pas sensiblement : *Notre Dame de Paris*, le roman de Victor Hugo (codé par ABU), fait 5% de plus en utf-8 qu'en latin-1. En revanche les textes Ascii sont déjà en utf-8 (au « détail » près de l'apostrophe ! voir section 6.1.1)

¹²Les fichiers `.sty` étant codés généralement en Ascii n'ont pas besoin d'être recodés (voir note précédente). Il existe de nombreuses façons de « recoder » des

Puisque utf-8 est implicite dans les textes (Lua)L^AT_EX, la ligne `\usepackage[utf8]{inputenc}` est inutile non seulement dans les programmes à compiler avec LuaL^AT_EX et X_YL^AT_EX, mais aussi avec L^AT_EX depuis les versions postérieures à avril 2018¹³.

À part les caractères réservés (au langage de commande T_EX, par exemple `\ & $ _ %`) on peut saisir directement les caractères qui ont une touche au clavier (en frappe directe ou composée) ; par exemple la touche A produit... un A. Mais, puisque les claviers standards proposent désormais des caractères Unicode, ceux-ci peuvent ainsi être saisis directement. Typiquement, sur mon Mac je peux taper les caractères « Ç, œ, æ, ß, £, €, °, ‰, §, ¶, ø, ∞, ÷, †, etc. » directement dans mon texte source.

Puisque les principales plateformes modernes sont en utf-8, il est possible de faire un copier depuis un autre document affiché sur l'écran en mode caractère (par exemple en pdf) puis un coller dans le texte source à l'écran.

Pour imprimer le symbole ☞, il suffit de le copier depuis un tableau Unicode et de le coller ici dans le texte.

fichiers (T_EX ou autres) :

— Le plus simple est sans doute de lire le fichier latin-1 avec un éditeur de T_EX (emacs, TeXShop, ...) et d'en demander une copie en utf-8.

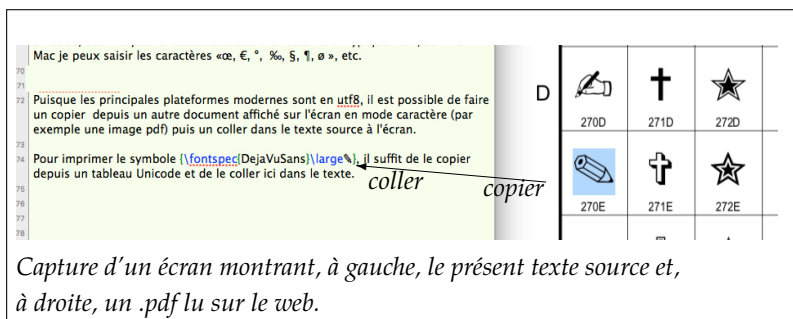
— Avec Unix, Linux, MacOS... : utiliser `iconv` (qui remplace `recode`) :

```
iconv -c -f LATIN1 -t UTF-8 source-latin1.tex > source-utf8.tex
```

avec l'option « `-f` » qui permet de forcer la conversion même s'il y a des caractères non latin-1 (ça existe souvent avec les espaces ou des choses comme ce saisis par des éditeurs pas très respectueux des normes, et surtout suite à des copier-coller).

— Avec Windows : il existe un équivalent à `iconv` pour Windows : `Liblconv`. On peut aussi utiliser `Powershell` etc. <à compléter?>

¹³Voir les *L^AT_EX News* n° 28, p.ex. à <https://www.latex-project.org/news/2018/04/10/issue28-of-latex2e-news-released/>.



Ceci ne marche bien sûr que si la fonte utilisée connaît le caractère copié.

Ce couper-coller peut aussi se faire sur des textes plus longs. Ainsi ayant vu sur le web un texte chinois (où l'on nous apprend que ça veut dire quelque chose comme « Si tu as de bons résultats, tes parents seront très fiers »), on peut le copier et le coller dans ces lignes :

```
\centerline{\fontspec{Arial Unicode MS}
你成绩好,父母脸上也不知有多好看。
}
```

ce qui imprime évidemment :

你成绩好,父母脸上也不知有多好看。

Bien-sûr, ce n'est pas la bonne façon de saisir du chinois, mais à l'occasion cette méthode peut s'avérer utile à quelqu'un qui ne connaît pas cette langue et veut simplement introduire quelques mots dans un texte français... Notons qu'ici on n'a pas pu utiliser la fonte DejaVuSans qui n'a pas les caractères CJK.

4. À PROPOS DES FONTES UNICODE

Rappelons encore une fois que toutes les fontes dites Unicode (utilisant donc le codage Unicode) n'ont pas la totalité des caractères prévus par Unicode. Loin de là... Mais, les caractères européens et beaucoup des caractères spéciaux sont en général dans les fontes « courantes » comme TeXGyrePagella, DejaVu, Times, Lucida, Arial, etc. Certaines fontes tentent d'être complètes, comme Arial Unicode MS

(50 000 glyphes) et Bitstream Cyberbit (30 000 glyphes). On trouve facilement sur le web des listes de fontes avec leur couverture¹⁴.

L'annexe (page 31) donne un programme listant les caractères présents dans une fonte Unicode donnée.

4.1. COMBINAISON DE CARACTÈRES

Dès les débuts de T_EX, la combinaison de touches permet de désigner de nombreux caractères : la capitale accentuée À peut être obtenue par la combinaison des touches `\`A` et le signe `&` par `\&`, etc. Ceci est bien sûr toujours utilisable avec `fontspec` ; exemples :

Code	Sortie
<code>\`œ</code>	œ
<code>{\fontspec{DejaVu Sans} \`\char"27B3}</code>	»

Mais le texte source étant désormais en utf-8, on peut utiliser les propriétés combinatoires d'Unicode et en particulier le positionnement de diacritiques. Par exemple, en utilisant le caractère U+034C (DIACRITIQUE PRESQUE ÉGAL À EN CHEF – caractère utilisé dans l'alphabet phonétique) et celui U+035B (DIACRITIQUE ZIGZAG EN CHEF – abréviation latine encore utilisée dans les imprimés du xvi^e siècle pour abréger *am, ar...*), on peut écrire l'exemple suivant (notons que le diacritique se met après la lettre le supportant) :

Code	Sortie
<code>\fontspec{Arial} a\char"034C, q\char"035B</code>	ã, ħ

Les mots des divers alphabets européens peuvent donc souvent s'écrire de plusieurs façons. Pour les passionnés de football, le nom du joueur croate Mateo Kovačić peut se composer comme suit :

	Code	Sortie
1	<code>Kovačić</code>	Kovačić
2	<code>Kova\ v ci\ 'c</code>	Kovačić
3	<code>Kova\char"010Di\char"0107</code>	Kovačić
4	<code>Kovac\char"030Cic\char"0301</code>	Kovačić

Ligne :

¹⁴P.ex. https://en.wikipedia.org/wiki/List_of_typefaces

1. on a fait un couper-coller des caractères Unicode, č et ć (ou on a utilisé un clavier croate où ces signes sont présents!),
2. on utilise la composition de T_EX (ce n'aurait pas été possible pour l'exemple du ě vu juste ci-dessus),
3. on utilise les numéros de code des caractères U+010D (č) et U+0107 (ć),
4. on utilise les flottants U+030C (DIACRITIQUE CARON OU HACEK, « ˇ ») et U+0301 (DIACRITIQUE ACCENT AIGU, « ´ ») qui ne chassent pas.

4.2. CARACTÈRES DES ZONES PRIVÉES D'UNICODE

Enfin, Unicode a prévu des zones d'usage privé (de U+E000 à U+F8FF notamment) réservées aux programmeurs et aux utilisateurs qui ont besoin d'un jeu de caractères particulier. On peut y mettre des caractères « non standards ». Il faut bien sûr qu'il y ait entente entre les utilisateurs de ces signes et disposer d'une fonte ad hoc.

C'est ainsi que le projet MUF1¹⁵ de codage de manuscrits anciens a défini une liste de caractères en complément à ceux d'Unicode, notamment pour les médiévistes du Nord de l'Europe, comme M+E735 ꝥ LATIN SMALL LIGATURE THORN AND LONG S WITH STROKE

M+F160 ʔ PUNCTUS INTERROGATIVUS

M+EED7 ꝰ LATIN SMALL LIGATURE PP WITH FLOURISH

Le projet PICA donne une liste des caractères français et latins anciens présents dans Unicode+Mufi¹⁶. Ces caractères fonctionnent comme ceux d'Unicode. Par exemple, les ouvrages de médecine ou de pharmacopée du xvi^e au xix^e siècle (voire aujourd'hui encore) utilisaient des poids tels que le dragme, le scrupule ou l'once. Tous ces caractères sont dans Unicode, sauf le *dragme* qui, sous prétexte qu'il ressemble à un 3, en a été exclu. Pour des raisons évidentes de distinction, des paléographes ont demandé à MUF1 d'inclure ce signe dans une zone privée, ce qui a été fait (en F2E6). La fonte Cardo (voir note 15) utilisée ici permet alors d'imprimer ce caractère « privé »

¹⁵ *Medieval Unicode Font Initiative*, voir <http://www.mufi.info>. On y trouve la liste de ces codes privés et la fonte Cardo qui supporte ce codage.

¹⁶ Avec un ordre plus logique que celui plutôt aléatoire d'Unicode : <http://jacques-andre.fr/PICA/SIGMA-PICA.pdf>

```

\newcommand{\cardo}[1]{\fontspec{Cardo}
    \fontsize{40}{40}\selectfont \char"#1}
\centerline{\begin{tabular}{|cccc|}\hline
\cardo{2114} & \cardo{F2E6}& \cardo{2125}& \cardo{2108}\\
livre & dragme & once & scrupule\\\hline
\end{tabular}}

```

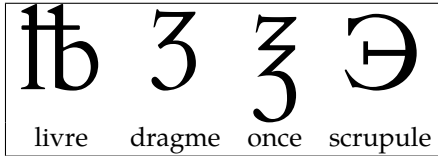


FIGURE 4.1. Appels de caractères normaux et de la zone privée (le dragme, en U+F2E6) d’Unicode

(voir figure 4.1), ce qui ne serait par exemple pas possible avec les fontes précédentes DejaVuSans ou Arial Unicode MS.

5. FONTES NON-UNICODE

5.1. FONTES CASSEAU

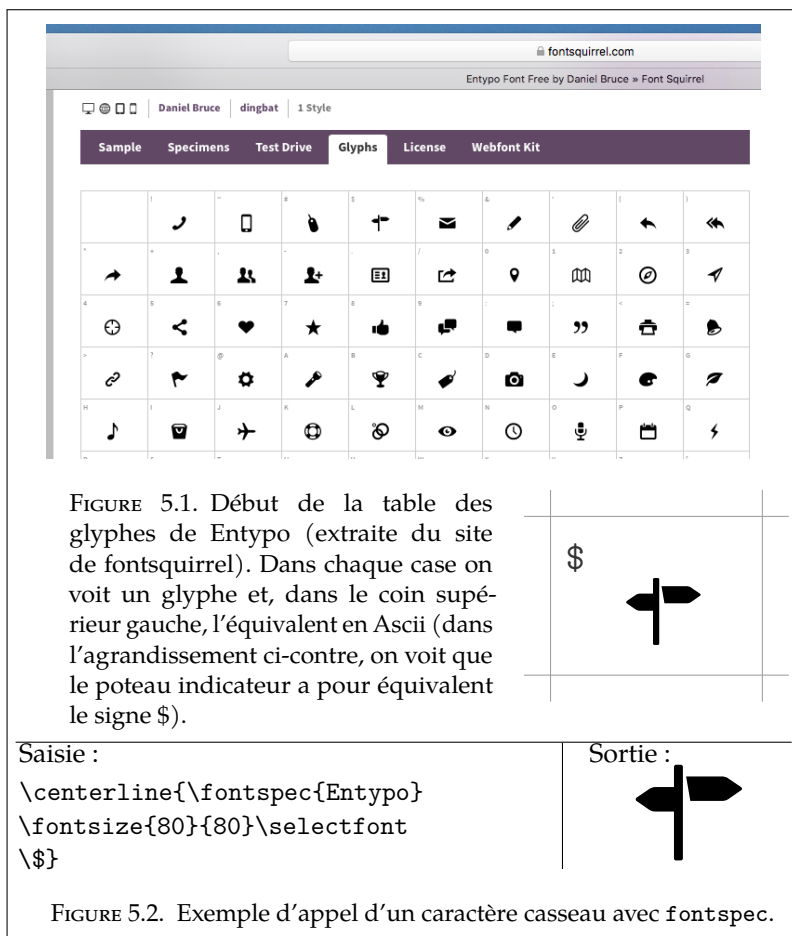
On sait que les fontes Unicode ont de plus en plus de dingbats et autres pictogrammes. Mais bien sûr pas tous et il faut encore recourir à des fontes casseaux pour de tels signes. C’est le cas de la fonte Entypo¹⁷. La table des glyphes (sur le site) montre qu’elle contient, à la place du signe dollar, un poteau indicateur qui nous intéresse. La figure 5.2 montre un exemple d’emploi de cette fonte (après avoir téléchargé Entypo.otf dans le répertoire courant).

Un autre emploi de Entypo, plus utile!, est donné en figure 5.3.

5.2. ORNEMENTS, BORDURES ET PAVAGES

Une classe particulièrement intéressante de casseaux correspond aux fontes spécialisées offrant des éléments permettant de construire des ornements, cadres, pavages, etc. Elles ne contiennent en général

¹⁷Téléchargeable librement depuis le site de Font Squirrel : <http://www.fontsquirrel.com/fonts/list/style/Dingbat/350>.



que quelques glyphes mais avec une métrique précise (par exemple des multiples du quart de cadratin) permettant des positionnements dans le plan avec d'éventuelles rotations. Unicode propose quelques-uns de ces signes – comme le plan U+25A0 dont par exemple ▲, ou, depuis la version 11, les *Ornamental Dingbats* de la rangée 1F650–1F67F, par exemple 🏹. Mais en nombre très limité et il faut souvent recourir à des fontes casseaux. Monotype, en particulier, a numérisé plusieurs de

Code source :

```
\begin{itemize}[label={\fontspec{Entypo}\huge Ä\,}]  
  \item Première indication  
  \item Une autre  
  \item Encore une autre  
\end{itemize}
```

Sortie :

- ▶▶ Première indication
- ▶▶ Une autre
- ▶▶ Encore une autre

FIGURE 5.3. Emploi de puces de liste depuis un casseau

ses fontes du temps du plomb¹⁸, dont notamment les séries *Arabesque ornaments* et *Rococo ornaments*. Une fois (achetées et) téléchargées, leur emploi à nouveau est très simple.

Prenons par exemple la fonte *RococoOrnamentsThreeMT* (où MT signifie Monotype). Faute de spécimen (comme en figure 5.1, mais voir figure 8.1), on essaye de voir quels sont les caractères présents en tapant par exemple

```
\fontspec{RococoOrnamentsThreeMT} AZERTY ABCDEF
```

qui donne `|\bullet\~\bullet|\bullet\~\bullet|`, ce qui est peu parlant. Mais en se rappelant que Monotype est d'origine américaine, on peut essayer avec le clavier QWERTY, comme en figure 5.4 qui montre que la répartition des ornements n'est pas faite au hasard. La figure 5.5 montre alors un emploi simple de ces ornements.

Les commandes de L^AT_EX restent évidemment valables pour ces caractères casseaux. Dans le cas des ornements notamment, il est très utile de connaître leurs propriétés métriques¹⁹. On peut donc utiliser

¹⁸Voir par exemple <http://jacques-andre.fr/faqtypo/orn/warde.html>. Max Caflisch notamment s'en était servi pour ses *Petits jeux avec des ornements* : <http://jacques-andre.fr/faqtypo/orn/jeux.pdf>.

¹⁹On peut aussi les avoir par un programme lua, en utilisant les propriétés données par OpenType. Voir <quelle ref. ?> et page 32.

```

{\fontspec
  {RococoOrnamentsThreeMT}
  1 2 3 4
  q w e r
  a s d f
  z x c v
  }

```

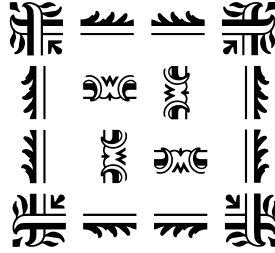


FIGURE 5.4. Les ornements sont souvent répartis de façon logique sur les claviers. Ici RococoOrnamentsThreeMT de Monotype.

Source :

```

{\fontspec
  {RococoOrnamentsThreeMT}
\begin{picture}(9em,5em)
\put(0,5em){1222w3334}
\put(0,4em){qNNNNNNNr}
\put(0,3em){eNNNNNNNs}
\put(0,2em){aNNNNNNNf}
\put(0,1em){zxxxdcccv}
\end{picture}}

```



FIGURE 5.5. Utilisation d'une fonte d'ornements pour faire un cadre

des commandes `\setto...` comme en figure 5.6²⁰ pour faire le tableau ci-dessous (dans le cas d'un corps 12, c.-à-d. avec 1 em=12pt). On y voit diverses espaces et que les valeurs des profondeurs montrent que souvent ces ornements sont sous la ligne de base (voir figure 5.7), mais c'est cohérent et une figure comme la 5.5 se fait « toute seule ».

²⁰On y utilise `\global` pour ne pas écrire les résultats avec la fonte Rococco... et pour ne pas avoir à l'appeler plusieurs fois.

Saisie

```
\newlength{\Machasse}\newlength{\Maprof}\newlength{\Mahaut}  
\fontspec{RococoOrnamentsThreeMT}\fontsize{12}{12}\selectfont  
\settowidth{\Machasse}{1}\global\Machasse=\Machasse  
\settoheight{\Maprof}{1}\global\Maprof=\Maprof  
\settoheight{\Mahaut}{1}\global\Mahaut=\Mahaut  
chasse=\the\Machasse  
profondeur=\the\Maprof  
hauteur=\the\Mahaut
```

Sortie

```
chasse=12.0pt  
profondeur=4.152pt  
hauteur=7.884pt
```

FIGURE 5.6. Calcul des métriques d'un caractère d'une fonte casseau

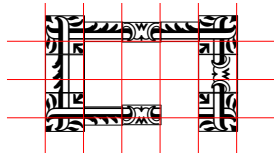


FIGURE 5.7. Les ornements peuvent être décalés (grâce à leur métrique) par rapport à la grille de la page (ici grille de 1em).

Clavier	Glyphe	Chasse	Hauteur	Profondeur
l		12.0pt	7.884pt	4.152pt
e		12.0pt	7.848pt	4.176pt
w		12.0pt	5.82pt	0.144pt
d		12.0pt	3.876pt	2.076pt
N	—	12.0pt	0.0pt	0.0pt
n	—	6.0pt	0.0pt	0.0pt
B	—	3.996pt	0.0pt	0.0pt
b	—	3.0pt	0.0pt	0.0pt


6. FONTES DE TITRAGE

Comme leur nom l'indique, les *fontes de titrage* permettent de composer des titres²¹. Non pas, en général, des titres de chapitres, sections, etc. (nous y revenons en 7) mais plutôt des couvertures, des affiches ou papillons (*flyers* en bon français...), etc. Ces caractères servent à composer des textes « à voir » – à accrocher l'œil, par leur graisse, leur (in-)esthétique, leur fantaisie, etc. – plus que « à lire » (contrairement aux caractères des fontes de labeur). Ces fontes ne sont composées que des quelques lettres capitales (parfois accentuées), rarement des minuscules correspondantes (si non, en général lorsqu'on tape par exemple un *a* on obtient un *A*), éventuellement les chiffres et très peu d'autres signes. Ce qui est intéressant, c'est que ces fontes sont auto-suffisantes : pas de ligatures (normalement les capitales ne sont pas ligaturables), pas de déclinaisons (italique, gras, etc.), pas de variantes... Il suffit donc de les télécharger et de les utiliser telles que. Il n'y a donc pas plus simple...

Prenons par exemple la fonte `Bolton_Drop_Caps.ttf`²² » que nous avons téléchargée dans notre répertoire courant; on peut alors composer le titre suivant (où l'on regrette que l'espace ne soit pas noir lui aussi...):

²¹Historiquement, dès le xvi^e siècle, ces fontes étaient fondues de façon que l'œil des capitales occupe toute la surface du type, c'est-à-dire qu'il n'y avait pas d'espace sous la ligne de base. Cette technique est rarement conservée dans les fontes de titrage numériques.

²²C'est typiquement le genre de fonte que l'on trouve sur le web, ici à http://www.acefreefonts.com/font-10455-Bolton_Drop_Caps.html. Il s'agit d'une fonte gratuite. En général il faut se méfier de ces fontes gratuites qui sont parfois de mauvaise qualité, qui peuvent parfois endommager le système et surtout qui ignorent souvent les problèmes de propriété intellectuelle et de copyright. Mais certains sites pourtant commerciaux (pour une liste, voir par exemple <http://www.graphiste-libre.com/15-sites-pour-trouver-des-polices-gratuites/>) en proposent parfois et ce sont surtout celles-là que nous utilisons ici.

<pre>{\fontspec{% Bolton_Drop_Caps.ttf} \fontsize{20}{20} \selectfont MON TITRE }</pre>	
---	--

De même, avec la fonte « chapitre²³ » qui montre que certaines fontes de titrage ont bien des capitales accentuées :

<pre>{\fontspec{chapitre}\huge UN BEL ÉTÉ }</pre>	
---	--

Quant à la lettrine du début de cet article (page 3), elle a été obtenue par

```
\renewcommand*{\LettrineFontHook}{\fontspec{Spring}}
\lettrine[lines=3,loversize=0.1]{B}{ien} souvent
```

toujours en ayant téléchargé la fonte correspondante depuis le web.

6.1. MODIFICATION DES CARACTÉRISTIQUES DES FONTES AVEC FONTSPEC

Ces fontes de titrage sont un cas limite des fontes casseaux dans la mesure où elles sont plutôt des fontes de labeur dont l'alphabet est limité essentiellement aux capitales. D'ailleurs certaines familles de fonte de qualité ont en plus de l'italique, du gras, etc. une variante *titling*. C'est le cas de AGaramond-Titling, BemboTitlingMT, CaslonTitlingMT, PlantinTitlingMT, Utopia-TitlingCapitals, etc. Ce côté labeur nous oblige à dire deux mots ici sur les options proposées par fontspec pour tenir compte des caractéristiques des fontes TrueType et surtout OpenType.

²³Cette fonte *chapitre* est au catalogue de la fonderie typographies.fr créée par deux anciens de l'École Estienne, Jonathan Fabreguettes-Perez et Laurent Bourcellier; cette fonte a été primée en 2014 par le Type Director Club. Cette fois, il s'agit d'une fonte payante (voir <http://www.typographies.fr/N/chapitre/chapitre.html>) comme beaucoup de fontes professionnelles.

6.1.1. L'option `Ligatures=TeX`

Prenons le cas du Garamond d'Adobe²⁴ qui comprend un fichier `AGaramond-Titling.otf`. Ni `fontspec`, ni `TeX` ne permettent de considérer cette variante au même titre que les italiques, gras voire petites capitales²⁵, aussi doit-on utiliser ces caractères de titrage comme faisant partie d'une fonte isolée.

Reprenons d'abord l'exemple d'« un bel été » (ci-dessus) mais en prenant le titre d'un thriller suédois connu et en le composant en Garamont. On écrit donc naturellement

<pre>{\fontspec {AGaramond-titling} \huge FIN D'ÉTÉ }</pre>	FIN D'ÉTÉ
---	-----------

Mais ça ne va pas... L'apostrophe est ce que d'aucuns appellent « la chiure de mouche », c'est-à-dire un trait droit. La faute est à Unicode²⁶ qui a confondu glyphe et caractère et a décidé que le caractère apostrophe n'avait désormais plus le code U+0027 (comme en Ascii et Latin-1) mais U+2019, se mettant de plus en contradiction avec son propre principe de compatibilité... Ce ne serait pas trop gênant si l'apostrophe n'était aussi fréquente que beaucoup de lettres (il y a plus d'apostrophes que de « b » dans un texte français) et qu'il faudrait donc recoder tous nos textes en remplaçant « ' » par « ´ » ou par `\char"2019`. Heureusement, depuis longtemps, `TeX` a distingué codage en entrée du caractère du codage du caractère imprimé en

²⁴Elle a été dessinée par Robert Slimbach pour la famille *Adobe Originals historical revivals*, avec Adobe Jenson Pro et Adobe Caslon Pro. Voir <https://www.myfonts.com/fonts/adobe/garamond/>. Ces fontes ne sont pas publiques et ne sont donc pas dans `TeXLive`.

²⁵La commande suivante est impossible
`\setmainfont{AGaramond}[Extension = .otf, UprightFont = *-regular , BoldFont = *-bold , ... TitleFont = *-titling]}` % commande inconnue.
Par ailleurs demander la propriété `RawFeature=+titl` ne donnerait pas le résultat attendu.

²⁶Voir mon article dans *Graphé* n° 39 à <http://jacques-andre.fr/japublis/Apostrophe-graphe39.pdf>.

sortie grâce à ses conventions de saisie, par exemple « ' » pour obtenir « ’ », « -- --- » pour avoir « — » ou « `` ' ' » pour obtenir « “ ” ». Ces conventions étaient traitées par le routines d’inputenc. Mais fontspec, lui, traite ces mêmes conventions en utilisant des propriétés des fontes. Celles-ci sont ainsi prises en compte lorsque l’on écrit [Ligatures=TeX]²⁷ en ouvrant une fonte par \fontspec. D’où la version corrigée de notre exemple :

<pre>{\fontspec[Ligatures=TeX] {AGaramond-titling} \huge FIN D'ÉTÉ }</pre>	<h1>FIN D'ÉTÉ</h1>
--	--------------------

Il devrait en être de même avec \setmainfont et \setsansfont. Mais, depuis la version 2015 de T_EXLive, le fichier fontspec.cfg contient :

```
\defaultfontfeatures[\rmfamily,\sffamily]{Ligatures=TeX}
```

c.-à-d. que TeX Ligatures est mis par défaut pour les fontes chargées par \setmainfont ou par \setsansfont. La figure 6.1 donne un exemple²⁸ montrant les quatre cas possibles. On y indique aussi, sans plus de détails²⁹, comment supprimer les conventions de T_EX.

6.1.2. Ligatures communes

On remarque sur les exemples de la figure 6.1 que les ligatures « fi » et « ff » ne sont pas affectées par l’absence de la commande Ligatures=TeX (l’emploi du mot *ligatures* est plutôt curieux, voire trompeur!) : en fait ce sont les fontes elles-mêmes qui ont, par défaut, l’activation de ces ligatures communes (avec aussi « ffi, ffl, » etc. – voir le manuel fontspec pour plus de détails ainsi que pour le traitement des autres ligatures). Mais on peut supprimer cette option, même localement si on veut. Exemple (comparez les deux « fi ») :

²⁷Le mot *Ligatures* est trompeur...

²⁸Utilisant la fonte de la labeur Erewhon. Il s’agit d’une extension de Heuristica, elle même variante de Utopia. Voir [2]. Elle est dans T_EXLive, ce qui ne change rien à cet exemple.

²⁹Disons cependant que l’écriture qu’on attendrait, [Ligatures=NoTeX], n’est pas valable.


<pre> \documentclass{article} \usepackage{fontspec} \begin{document} {\fontspec{erewhon} 1. --- l'air du ``fiffre'' -- } {\fontspec[Ligatures=TeX]{erewhon} 2. --- l'air du ``fiffre'' -- } {\setmainfont{erewhon} 3. --- l'air du ``fiffre'' -- } {\setmainfont[Ligatures=TeX]{erewhon} 4. --- l'air du ``fiffre'' -- } {\setmainfont[RawFeature=-tlig]{erewhon} 5. --- l'air du ``fiffre'' -- } \end{document} </pre>	<pre> 1. --- l'air du ``fiffre'' -- 2. — l'air du “fiffre” – 3. — l'air du “fiffre” – 4. — l'air du “fiffre” – 5. --- l'air du ``fiffre'' -- </pre>
--	---

FIGURE 6.1. Exemples d'utilisation de [Ligatures=TeX].


<pre> {\fontspec{erewhon} La fin de l'air du {\addfontfeatures {Ligatures=NoCommon} fiffre --}} </pre>	<p>La fin de l'air du fiffre --</p>
--	-------------------------------------

6.1.3. Crénage

Revenons à nos fontes de titrage et considérons l'exemple suivant d'une couverture de roman dont on veut composer le titre avec une fonte plutôt graphique³⁰ :

<pre>{\fontspec[Ligatures=TeX] {Sofachrome-rg} \begin{center} À L'AVAL\DE L'IF \end{center}}</pre>	
--	--

Cette fois on a commandé la bonne apostrophe mais on voit que si elle est de façon heureuse bien crénée avec le L de L'IF (elle est au dessus de la barre du L ce qui diminue le blanc entre le L et le I), elle pose un problème dans L'A : elle est bien crénée avec le L, mais aussi avec le A, ce dernier venant alors chevaucher la barre du L (on remarquera le même phénomène avec le V qui est créné avec le A le précédant, mais aussi avec celui le suivant ; mais cette fois c'est plutôt bien!). Ici encore, fontspec permet de supprimer localement le crénage (ici pour le L de L'AVAL, ce qui ne supprime pas le crénage apostrophe-A), et on peut écrire le texte suivant :

<pre>{\fontspec[Ligatures=TeX] {Sofachrome-rg} \begin{center} À {\addfontfeatures {Kerning=Off}L}'AVAL\ DE L'IF \end{center}}</pre>	
---	---

³⁰Fonte sofachrome de Ray Larabie que l'on a trouvée sur le web (à <http://typodermicfonts.com/alphabetical/page/16/>) et qu'on utilise ici avec l'autorisation de son auteur.

7. CHANGER LA FONTE DES TITRES

Que ce soit avec L^AT_EX ou LuaL^AT_EX, une même fonte est utilisée, pour des questions évidentes d’homogénéité typographique, pour tout un document, avec des variations automatiques de taille, graisse, etc. pour les titres de sectionnement, titres courants, etc. selon les protocoles des classes standards (comme article, book,...) ou non. Cette fonte globale est définie par défaut ou par l’utilisateur lui-même. La figure 7.1 montre deux programmes minima utilisant la version publique du Times Roman³¹.

Avec L ^A T _E X et NFSS	Avec LuaL ^A T _E X et fontspec
<pre>\documentclass{article} \usepackage{tgtermes} \begin{document} \section{Écrire en \emph{Times}} abc def \emph{abc defi gh} \end{document}</pre>	<pre>\documentclass{article} \usepackage{fontspec} \setmainfont{TeXGyreTermes} \begin{document} \section{Écrire en \textit{Times}} abc def \emph{abc defi gh} \end{document}</pre>

1 Écrire en *Times*

abc def *abc defi gh*

FIGURE 7.1. Deux façons de choisir du Times pour tout un document T_EX et le résultat

En revanche, il faut bien voir que la gestion d’une fonte (notamment le choix des caractères eux-mêmes : graisse, taille, forme, etc.) relève de règles typographiques et c’est justement à ça que servent les classes article, memoir, book, etc. Nous nous adressons ici aux utilisateurs de ces classes et non à leur développeurs et ce n’est donc pas la place de faire tout un cours sur la gestion de fontes <qui soit dit

³¹Que ce soit avec NFSS ou avec fontspec, la dénomination des fontes reste toujours un sujet délicat!

en passant serait à re-faire au vu de `fontspec!`>. Contentons donc nous de demander aux auteurs de faire confiance à ces classes, ou d'en rechercher une qui corresponde à un besoin particulier³².

MAIS il peut quand même arriver des cas où on a besoin d'utiliser pour un titre d'une fonte différente de celle prévue par la classe qu'on utilise. Pour ces cas vraiment isolés, on peut utiliser `fontspec` tout comme pour un caractère casseau...

Reprenons l'exemple de la figure 7.1. On peut aisément le modifier comme en figure 7.2. On y utilise des fontes déjà vues plus haut et la variante de TeXGyre du Helvetica. On voit (notamment avec les ---) qu'il est utile d'entrer un minimum dans les paramétrages de fontes comme on l'avait déjà fait en section 6, et que si une fonte n'a pas tous les caractères (ce qui est le cas de Bolton), `fontspec` ne va pas les inventer.

Si maintenant on veut que tous les titres de section soient dans une fonte différente de celle du texte courant, ça devient plus compliqué, mais encore faisable par un non-expert. Il faut modifier la définition de la commande `\section` dans le prologue. Pour la classe de notre exemple 7.2, `article`³³, ça peut se faire comme suit :

```
\makeatletter
\renewcommand\section{\@startsection {section}{1}{\z@}%
                                   {-3.5ex \@plus -1ex \@minus -.2ex}%
                                   {2.3ex \@plus .2ex}%
                                   {\fontspec{TeXGyreHeros}[Ligatures=TeX]
                                   \normalfont\Large\bfseries}}
\makeatother
```

Tous les titres de section seront alors en Helvetica.

Mais il faudrait sans doute faire de même pour `\subsection`, `\subsubsection`, `\paragraph`, sans oublier les commandes pour la

³²Toutefois il existe, sur CTAN, des packages comme `titlesec` et `sectsty` qui donne accès à la définition de ces sectionnements grâce à des macros comme `\allsectionsfont`.

³³Le source de cette classe se trouve (cette année) normalement dans `/usr/local/texlive/2019/texmf-dist/tex/latex/base/article.cls` où on trouve la définition `\newcommand\section...`

```

\documentclass{article}
\usepackage{fontspec}
\setmainfont{TeXGyreTermes}
\begin{document}
\section{Écrire en \textit{Times}}
Ceci est du Times
\section{\fontspec{TeXGyreHeros}
  Titre en \textit{Helvetica} ---}
Ceci est du Times
\section{\fontspec{TeXGyreHeros}[Ligatures=TeX]
  Titre en \textit{Helvetica} ---}
Ceci est du Times
\section{\fontspec{Bolton_Drop_Caps.ttf}[Ligatures=TeX]
  Fièrè de \emph{Bolton} ---}
et ce texte courant est toujours en Times...
\end{document}

```

Sortie :

1 *Écrire en Times*

Ceci est du Times

2 *Titre en Helvetica ---*

Ceci est du Times

3 *Titre en Helvetica —*

Ceci est du Times

4 **F I R È D E B O L T O N** □□□

et ce texte courant est toujours en Times...

FIGURE 7.2. Utilisation de fontspec (version v2.7h du 2020/02/03) pour modifier localement la fonte d'un titre.

table des matières, celles pour les titres courants, etc. Ce qui confirme donc bien ce que nous disions que ceci n'est pas une modification locale, mais relève de la réécriture d'une classe entière.

8. EN GUISE DE CONCLUSION

On a voulu montrer par ces exemples d'une part que fontspec permet d'imprimer sans conversion un caractère isolé d'une fonte non- \TeX et, d'autre part, que si fontspec permet d'accéder à la typographie fine³⁴, c'est en général inutile pour les dingbats et autres signes spéciaux... d'où l'intérêt de la commande brute `\fontspec{nom de fichier}`.

— **F I N** —

³⁴Le manuel [5] donne toutes les commandes en détail ; on a vu que même pour des modifications locales de la fonte d'un titre, on est quasi-obligé d'utiliser ces finesses.

BIBLIOGRAPHIE

- [1] Patrick ANDRIES et col., *Liste des noms Unicode 10.0 et ISO/CEI 10646:2017 annotés*, Hapax, 2017. <http://hapax.qc.ca/ListeNoms-10.0.0.txt>
- [2] Palle JØRGENSEN, *The L^AT_EX Font Catalogue*, <http://www.tug.dk/FontCatalogue/about.html>
- [3] LIANTZE Lim (Overleaf), *\fontspec All the Fonts!*, décembre 2016. <https://www.overleaf.com/articles/slash-fontspec-all-the-fonts/qnsxyhrjsgs#.W3nJ0S3p0S4>
- [4] Scott PAKIN, *The Comprehensive L^AT_EX Symbol List*, CTAN ou <http://faculty.cse.tamu.edu/hlee/csce222/LaTeXSymbols-letter.pdf>
- [5] Will ROBERTSON, *The fontspec package – Font selection for X_ƎL^AT_EX and LuaL^AT_EX*, with contributions by Khaled Hosny, Philipp Gesang, Joseph Wright, and others; version v2.7h, 2020/02/03. <http://mirrors.ircam.fr/pub/CTAN/macros/latex/contrib/fontspec/fontspec.pdf>

ANNEXE : INVENTAIRE DES CARACTÈRES D’UNE FONTE

Il y avait en T_EX classique une façon simple de faire le tableau des caractères d’une fonte : `nffsfonts`. Il n’existe pas l’équivalent pour les fontes Unicode utilisables avec `fontspec`.

On peut voir l’ensemble des caractères présents dans une fonte Unicode (ou autre d’ailleurs) en l’ouvrant avec un gestionnaire de fonte comme `fontforge`; d’une part en regardant les glyphes affichés avec leur numéro, d’autre part en suivant le lien `Element/Font Info/Unicode Ranges`.

Il m’a été donné d’avoir besoin d’écrire un programme LuaL^AT_EX qui fait cet inventaire et je le joins à cet article. Il est relativement simple³⁵ : il regarde tous les caractères d’une fonte, numéro par numéro dans un intervalle donné, et si ce caractère existe (si il y a un glyphe correspondant qui ne soit pas vide) il imprime son numéro, son glyphe et le glyphe qu’il aurait dans une autre fonte (en général il faut qu’elle-même soit une fonte complète) au cas où les glyphes soient différents des glyphes Unicode attendus (on a vu que c’est souvent le cas des casseaux).

Il est donné en page 32 et on montre, en page 33, ce qu’il sort pour la fonte `RococoOrnamentsThreeMT` que nous avons utilisée plus haut.

³⁵Mais je tiens à remercier René Fritz et Arthur Reutenauer pour leurs conseils sur certains points délicats !

Programme faisant l'inventaire des caractères de la fonte \MaFonte entre les codes debut et fin (valeurs à mettre à jour).

```

%!TEX encoding = UTF-8 Unicode
% !TEX TS-program = LuaLaTeX
\documentclass[10pt]{article} % inventaires des car d'une fonte
\usepackage[a4paper,textwidth=15.75cm,%
    textheight=23.4cm,heightrounded]{geometry}
\newcommand{\MaFonte}{RococoOrnamentsThreeMT}% <= modifier
\newcommand{\FonteUnicode}{Arial Unicode MS}% <= modifier
\newcommand{\debut}{"O"% <= modifier
\newcommand{\fin}{"FF"% <= modifier
\usepackage{fontspec}
\usepackage{ifthen,xifthen,calc}
\usepackage{multido}\usepackage{multicol}
\usepackage{fmtcount}
\newcounter{ctr}\newcounter{nb}
\newcommand{\tableglyphes}[3]{%
Codes hexadécimaux des caractères de #1
\setcounter{nb}{#3-#2+1}%
\begin{multicols}{5}%
\newfontfamily\mafonte{#1}
\multido{\ia=#2+1}{\thenb}{\setcounter{ctr}{\ia}%
\mafonte
\ifthenelse{%
\directlua{
    local f = font.getfont(font.current())
    local chars = f.characters
    if chars[\ia] then
        tex.print('\string\\boolean{false}') %
    else
        tex.print('\string\\boolean{true}')
    end
}}{\texttt{\Hexadecimal{ctr}\mbox{ = \char\ia = }}%
{\mafonte\parbox[b][15pt]{15pt}{\strut\Large\char\ia}}\}}
\end{multicols}}%fin table glyphes
\parindent0pt
\setmonofont{\FonteUnicode}
\begin{document}
\tableglyphes{\MaFonte}{\debut}{\fin}
\end{document}

```


Codes hexadécimaux des caractères de RococoOrnamentsThreeMT






























































20 = =	33 = 3 = 	46 = F = 	59 = Y = •	6D = m = •
21 = ! = 	34 = 4 = 	47 = G = •	5A = Z = •	6E = n =
22 = " = 	35 = 5 = •	48 = H = 	5B = [= •	6F = o = 
23 = # = 	36 = 6 = •	49 = I = •	5D =] = 	70 = p = 
24 = \$ = 	37 = 7 = 	4A = J = 	5E = ^ = •	71 = q = 
25 = % = •	38 = 8 = 	4B = K = 	5F = _ = 	72 = r = 
26 = & = 	39 = 9 = 	4C = L = 	61 = a = 	73 = s = 
27 = ' = 	3A = : = 	4D = M = •	62 = b =	74 = t = •
28 = (= 	3B = ; = 	4E = N =	63 = c = 	75 = u = 
29 =) = 	3C = < = •	4F = O = 	64 = d = 	76 = v = 
2A = * = 	3D = = = 	50 = P = 	65 = e = 	77 = w = 
2B = + = 	3E = > = •	51 = Q = 	66 = f = 	78 = x = 
2C = , = •	3F = ? = •	52 = R = 	67 = g = •	79 = y = •
2D = - = 	40 = @ = 	53 = S = 	68 = h = 	7A = z = 
2E = . = •	41 = A = 	54 = T = •	69 = i = •	7B = { = •
2F = / = •	42 = B =	55 = U = 	6A = j = 	7D = } = 
30 = 0 = 	43 = C = •	56 = V = •	6B = k = 	A3 = £ = 
31 = 1 = 	44 = D = 	57 = W = 	6C = l = 	AD = =
32 = 2 = 	45 = E = 	58 = X = •		

FIGURE 8.1. Sortie du programme de la page précédente

REMERCIEMENTS

Je tiens à remercier tous ceux qui m'ont aidé à écrire ce texte ou à en corriger des versions provisoires, notamment Daniel Flipo, René Fritz, Arthur Reutenauer... À suivre

« Jacques ANDRÉ
Rennes, France
Jacques.Andre35@gmail.com