

# Extension garamondx\*

Michael Sharpe

18 novembre 2012

## I Introduction

Cette extension ajoute à ugm des fonctionnalités autrefois appelées *expert*, d'où le x. Les polices ugm, (URW)+ + GaramondNo8, ne sont pas libres au sens de la GNU, mais sont mis à disposition sous l'AFPL (*Aladdin Free Public License*), qui empêche leur diffusion au sein de T<sub>E</sub>XLive. Elles peuvent être téléchargées à l'aide du script `getnonfreefonts` qui fait partie de T<sub>E</sub>XLive. Les instructions d'installation sont disponibles sur,  
<http://tug.org/fonts/getnonfreefonts/>

Les polices de cette extension dérivent en définitive des polices ugm, et sont donc également soumises à la même licence AFPL dont les détails précis sont énoncés sur,  
<http://www.artifex.com/downloads/doc/Public.htm>

En termes généraux, la licence permet l'utilisation illimitée des polices par n'importe qui, mais ne permet pas un usage commercial pour leur distribution. Elle contraint également ceux qui modifient les polices à les délivrer sous la même licence, et les oblige à fournir des informations sur la nature des changements et leur identité.

Dans les polices ugm du CTAN, il manque :

- un ensemble complet de ligatures pour le f (`f_f`, `f_f_i` et `f_f_l` sont manquantes) ;
- les petites capitales ;
- les chiffres elzéviens.

Les glyphes eux-mêmes sont très proches de ceux de la distribution d'Adobe Stempel Garamond, qui a de nombreux admirateurs, mais ils n'ont pas, eux non plus, les ligatures du f. Le but ici est donc de créer une extension qui fournit ces fonctionnalités manquantes qui devraient, à mon avis, faire obligatoirement partie de toutes les extensions modernes de L<sup>A</sup>T<sub>E</sub>X.

Dans cette distribution, il manque dans le codage T1 le glyphe `perthousandzero`, qui est rarement présent dans les polices PostScript, et n'est presque jamais nécessaire dans les extensions L<sup>A</sup>T<sub>E</sub>X.

## 2 Un peu d'histoire

Contrairement à la plupart des autres polices dont le nom renferme le terme Garamond, les glyphes de cette police sont en fait des rendus numériques des polices effectivement conçues par

---

\*Traduit par René Fritz le 4 décembre 2012.

Claude Garamond dans le milieu du  $xvi^e$  siècle — la plupart des autres polices Garamond sont plus proches de polices conçues par Jean Jannon quelques années plus tard. La société Stempel possédait le spécimen à partir duquel ils ont conçu les moulages métalliques des polices dans les années 20. Les premiers rendus numériques comprennent ceux de Bitstream sous le nom OriginalGaramond et ceux d'Adobe Stempel Garamond, sous licence de Linotype. (Il semble que bon nombre des lacunes des polices conçues par Linotype étaient des artefacts dus aux limites des machines pour lesquelles les polices ont été conçues et n'ont, dans la plupart des cas, pas été corrigées.)

La dernière version (TrueType, non PostScript) officielle de (URW)+ + GaramondNo8 est disponible sur,

<http://ctan.org/tex-archive/support/ghostscript/AFPL/GhostPCL/urwfonts-8.71.tar.bz2>

sa collection de glyphes est plus étendue que celles des versions PostScript. En particulier, les ligatures du f sont présentes, ainsi que les glyphes Eng et eng qui dans le codage T1 sont obtenus respectivement par les commandes \NG et \ng.

À ma connaissance, il y a eu, assez récemment, deux tentatives pour retravailler ces polices. La première, sur laquelle ce travail est basé, est celle de Gael Varoquaux, disponible sur,

<http://gael-varoquaux.info/computers/garamond/index.html>

son extension ggm, n'étant pas distribuée sur le CTAN, ne semble pas avoir été largement diffusée.

La seconde est la distribution OpenType de Rogério Brito et Khaled Hosni disponible sur, <https://github.com/rbrito/urw-garamond>

Brito semble avoir fait un effort pour obtenir (URW)+ + afin de délivrer les polices sous une licence moins restrictive, qui ne semble pas avoir porté ses fruits. Leur projet s'adressait principalement les utilisateurs de Lua $\TeX$  et X $\LaTeX$ , et reste incomplet.

De l'extension ggm, j'ai gardé (a) une base de départ pour améliorer les métriques ; (b) le glyphe Q ornementé, mais pas par défaut.

### 3 Nouveautés

Les éléments les plus importants sont : (i) les polices nouvellement conçues pour les petites capitales normal, italique, gras et gras italique ; (ii) les chiffres elzéviens conçus pour chaque graisse ou et chaque style ; (iii) un ensemble complet de ligatures du f ; (iv) les macros pour personnaliser des chiffres ou le Q par défaut. Pour plus de détails sur les points (i) et (ii), voir la dernière section.

### 4 Options

Cette extension utilise le codage T1 qu'elle intègre en interne : il n'est donc pas nécessaire de l'appeler séparément. De même, l'extension `textcomp` est chargée automatiquement, vous donnant accès à de nombreux symboles absents dans le codage T1.

- L'option `scaled` peut être utilisée pour mettre toutes les polices à l'échelle indiquée par le nombre spécifié. Par exemple, `scaled=.9` dimensionne toutes les polices à 90 % de leur taille. Si vous entrez seulement l'option `scaled` sans valeur, la taille par défaut sera de 0.95, ce qui est la bonne mesure pour amener la hauteur des capitales (*Cap-height*) de

GaramondNo8 à environ 0.665 em ; ce qui est normal pour une police de texte, quoique inférieure à la hauteur d’x normale (*x-height*) : ce qui est typique des polices Garamond.

- Par défaut, l’extension utilise les chiffres alignés 0123456789 plutôt que les chiffres elzéviens 0123456789. L’option `osf` force le passage des chiffres au style ancien que je préfère, 0123456789, où le 1 ressemble au chiffre 1 aligné avec une hampe raccourcie, tandis que l’option `osfI` utilise les chiffres elzéviens traditionnels 0123456789, où le 1 ressemble à la lettre I avec une hampe raccourcie. Quelle que soit l’option utilisée,

- `\textlf{1}` produira le chiffre 1 aligné ;
- `\textosf{1}` produira mon style ancien préféré 1 ;
- `\textosfI{1}` produira la style ancien traditionnel 1.

- Par défaut, la lettre Q est dans la version traditionnelle de GaramondNo8. Elle peut être remplacée partout par la version ornementée à l’aide de l’option, `swashQ` qui vous donnera, par exemple, *Quoi* !

Que vous ayez précisé l’option `\swashQ` ou non, vous pouvez imprimer un Q ornementé dans la graisse et la forme courante en écrivant `\swashQ`. Par exemple, `\swashQ uash`, produira *Quash*.

## 4.1 Exemples

Voici les effets de quelques options :

```
\usepackage[scaled=.9,osf]{garamondx}% taille à 90 % avec mon style ancien
\usepackage[scaled,osf]{garamondx}% taille à 95 %, avec mon style ancien
\usepackage[osfI]{garamondx}% style traditionnel
\usepackage[osfI,swashQ]{garamondx}% style traditionnel & les Q ornementés
```

## 5 Chiffres supérieurs

Les versions TrueType de GaramondNo8 ont un ensemble complet des chiffres supérieurs, contrairement à leurs homologues PostScript. Seuls les glyphes des chiffres supérieurs de graisse ordinaire ont été copiés dans `NewG8-sups.pfb` et `NewG8-sups.afm` et introduits par un tfm nommé `NewG8-sups.tfm` qui peut être utilisé par l’extension `superiors` pour rendre les marqueurs de notes de bas de page ajustables. Consultez le `superiors-doc.pdf` (vous pouvez le trouver dans T<sub>E</sub>XLive en tapant `texdoc superiors` dans la fenêtre du Terminal.) L’appel le plus simple est :

```
\usepackage[supstfm=NewG8-sups]{superios}
```

## 6 Détails de mise en œuvre

### 6.1 Petites capitales

Les polices des petites capitales ont été créés à partir des capitales en utilisant FontForge pour les dimensionner de manière uniforme à 67 %, puis épaissir les pleins horizontaux et verticaux

à 130 %. Les résultats ont servi d'ébauche pour les réglages individuels de chaque glyphe. Avec FontForge, les pleins ont été corrigés de façon appropriée, nécessitant souvent un remaniement de la forme. Les résultats finaux sont la seule description possible de ces transformations. Enfin, les métriques appropriées ont été créées à l'aide FontForge, donnant résultats finaux. La forme verticale en graisse normale, a été retravaillé beaucoup plus que les autres graisses, et semble meilleure, à mon avis. Créer des petites capitales à partir de zéro exige un véritable effort pour obtenir les glyphes, les métriques et le bon crénage. Dans les deux formes droites normales et grasses, les glyphes standard accentués sont fournis, ainsi que des caractères spéciaux et des ligatures a\_e et o\_e et les glyphes lslash et oslash.

Les macros de petites capitales `\textsc` coopèrent avec `\textbf` et `\textit`, de sorte que vous pouvez utiliser, par exemple :

```
\textsc{Capitales et Petites Capitales}
```

pour produire **CAPITALES ET PETITES CAPITALES**,

```
\textit{\textsc{Capitales et Petites Capitales}}
```

pour produire *CAPITALES ET PETITES CAPITALES*,

```
\textbf{\textsc{Capitales et Petites Capitales}}
```

pour produire **CAPITALES ET PETITES CAPITALES**, et

```
\textbf{\textit{\textsc{Capitales et Petites Capitales}}}
```

pour produire *CAPITALES ET PETITES CAPITALES*.

## 6.2 Chiffres elzéviens

Les chiffres elzéviens ont été créés à partir des chiffres alignés existants, en réduisant les pleins du o et du 1 à une taille minuscule avec FontForge, et en abaissant les positions verticales des autres. Les formes ont ensuite été modifiées avec FontForge pour obtenir l'aspect plus traditionnel du style ancien. Les résultats finaux montrent les transformations impliquées.

## 7 Accord avec une extension math

L'extension `mathdesign` de Paul Pichaureau possède l'option `ugm` qui met le texte en `ugm` avec son extension `math` assortie. Pour utiliser cette extension `math` avec `garamondx` tapez :

```
\usepackage[ugm]{mathdesign}
\usepackage{garamondx}
```

Une autre possibilité est d'utiliser l'option `garamondx` de `newtxmath`, qui utilise les italiques `garamondx` capitales et minuscules, parfaitement adaptées aux maths, à la place des italiques Times utilisées par défaut. Cela nécessite la version 1.06 ou postérieure de l'extension `newtxmath`.

```

\usepackage{garamondx} % avec les chiffres alignés bons pour les maths
\usepackage[scaled=.84]{beramono} % bonne police machine à écrire
\usepackage{biolinum-type1} % une sans-serif
\usepackage[garamondx,cmbraces]{newtxmath}
\useosf % style des chiffres en osf pour le texte et non pour les maths

```

REMARQUE. – La dernière commande, ainsi que son homologue `\useosfI`, ne peuvent être utilisées que dans le préambule, et ne doivent jamais précéder `\usepackage{garamondx}`.

## 8 Lisence

Les polices de cette extension sont dérivées des polices (URW) + + GaramondNo8 qui ont été délivrées sous l'AFPL ; il en est donc de même pour ces polices. Les autres fichiers de support sont soumis à la *L<sup>A</sup>T<sub>E</sub>X Project Public License* ; voir sur le site,

<http://www.ctan.org/tex-archive/help/Catalogue/licenses.lppl.html>

pour les détails de cette licence.

Les modifications de cette extension et des polices décrites ci-dessus sont protégées par Copyright, Michael Sharpe, [msharpe@ucsd.edu](mailto:msharpe@ucsd.edu), le 3 octobre 2012.

### 8.1 Fichiers couverts par l'AFPL

```

NewG8-Bol.afm
NewG8-Bol.pfb
NewG8-Bol-SC.afm
NewG8-Bol-SC.pfb
NewG8-BolIta.afm
NewG8-BolIta.pfb
NewG8-BolIta-SC.afm
NewG8-BolIta-SC.pfb
NewG8-Ita-SC.afm
NewG8-Ita-SC.pfb
NewG8-Ita.afm
NewG8-Ita.pfb
newG8-0sf-bol.afm
newG8-0sf-bol.pfb
newG8-0sf-bolita.afm
newG8-0sf-bolita.pfb
newG8-0sf-ita.afm
newG8-0sf-ita.pfb
newG8-0sf-reg.afm
newG8-0sf-reg.pfb
NewG8-Reg-SC.afm
NewG8-Reg-SC.pfb
NewG8-Reg.afm
NewG8-Reg.pfb
NewG8-sups.afm
NewG8-sups.pfb

```