The garamondx package

Michael Sharpe

December 21, 2012

1 Overview

This package provides an extension of the ugm package, adding features that were once referred to as *expert*, whence the x. The ugm fonts, (URW)++ GaramondNo8, are not free in the sense of GNU but are made available under the AFPL (Aladdin Free Public License), which is restrictive enough to prevent their distribution as part of TEXLive. They may be downloaded using the getnonfreefonts script that used to be part of TEXLive. Instructions for installation are laid out at
http://tug.org/fonts/getnonfreefonts/
The fonts in this package are derived ultimately from the ugm fonts, and so are also subject to the same AFPL license, the precise details of which are spelled out at

http://www.artifex.com/downloads/doc/Public.htm

In broad terms, the license allows unlimited use of the fonts by anyone, but does not not permit any fee for their distribution. It also restricts those who modify the fonts to release them under the same license, and requires them to provide information about the changes and the identity of the modifier.

The ugm fonts on CTAN lack:

- a full set of f-ligatures (f_f, f_f_i and f_f_l are missing);
- small caps;
- old style figures.

The glyphs themselves are very close to those in Adobe's Stempel Garamond font package, which has many admirers, though they also lack the same f-ligatures. So, the goal here is to make a package which provides these missing features which should, in my opinion, be an essential part of any modern LATEX font package.

The only glyph missing from the T1 encoding in this distribution is perthousandzero, which is only rarely present in PostScript fonts, and is almost never required as part of LATEX packages.

2 Some History

Unlike most other fonts having Garamond as part of the name, the glyphs in this font are in fact digital renderings of fonts actually designed by Claude Garamond in the mid sixteenth century — most other Garamond fonts are closer to fonts designed by Jean Jannon some years later. The Stempel company

owned the specimen from which they designed metal castings of the fonts in the 1920's. Early digital renderings include those by Bitstream under the name OriginalGaramond, and Stempel Garamond from Adobe, licensed from LinoType. (It appears that many of the deficiencies of fonts designed by LinoType were artifacts of the limitations of the machines for which the fonts were designed, and have in most cases not been corrected.)

The latest version (TrueType, not PostScript) of the official (URW)++ GaramondNo8 is available from

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

which has a more extensive collection of glyphs than the PostScript versions. In particular, the f-ligatures are there, as well as the glyphs `Eng`, `eng` that are part of the T1 encoding under the names `Ng`, `ng`.

To my knowledge, there have been two fairly recent attempts to rework these fonts. The first, upon which this work is based, was by Gael Varoquaux, available at

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

His `ggm` package seems never to have been widely distributed, not having appeared on CTAN.

The second was an OpenType package by Rogério Brito and Khaled Hosny at

`https://github.com/rbrito/urw-garamond`

Brito seems to have made an effort to get (URW)++ to release the fonts under a less restrictive license, which does not appear to have been successful. Their project was aimed mainly towards users of LuaTeX and XeTeX, and remains incomplete.

What I kept from the `ggm` package was (a) a starting point for improved metrics; (b) the swash Q glyph, though not as the default Q.

3 New in this package

The most important items are (i) newly designed Small Cap fonts for Regular, Italic, Bold and Bold Italic; (ii) newly designed old style figures for each weight/style; (iii) a full set of f-ligatures; (iv) macros to allow customizations of the default figures and the default Q. For details of (i) and (ii), see the last section.

4 Package Options

The package uses T1 encoding—this is built into the package and need not be specified separately. Likewise, the `textcomp` package is loaded automatically, giving you access to many symbols not included in the T1 encoding.

- The option `scaled` may be used to scale all fonts by the specified number. Eg, `scaled=.9` scales all fonts to 90% of natural size. If you provide just the option `scaled` without a value, the default is `0.95`, which is about the correct scaling to bring the Cap-height of GaramondNo8 down to .665`em`, about normal for a text font, but with a smaller than normal x-height that is typical of Garamond fonts.

- By default, the package uses lining figures 0123456789 rather than oldstyle figures 0123456789. The option `osf` forces the figure style to a modified oldstyle that I prefer, 0123456789, where the 1 looks like a lining figure 1 with a shortened stem, while the option `osfI` uses the more traditional oldstyle figures 0123456789, where the 1 looks like the letter I with a shortened stem. No matter which option you use:
  - `\textlf{1}` produces the lining figure 1;
  - `\textosf{1}` produces my preferred oldstyle 1;
  - `\textosfI{1}` produces the traditional oldstyle 1.
- The default version of the letter Q is the traditional one from GaramondNo8. It may be replaced everywhere by the swash version via the option `swashQ`, which gives you, eg, Quoi?

  Whether or not you have specified the option `swashQ`, you may print a swash Q in the current weight and shape by writing `\swashQ` — eg,

  `\swashQ uash.`

  produces Quash.

4.1 Examples

The following show the effects of some options:

```
\usepackage[scaled=.9,osf]{garamondx}% scaled to 90%, my oldstyle
\usepackage[scaled,osf]{garamondx}% scaled to 95%, my oldstyle
\usepackage[osfI]{garamondx}% traditional oldstyle
\usepackage[osfI,swashQ]{garamondx}% traditional oldstyle, all Q rendered as swash Q
```

5 Superior figures

The TrueType versions of GaramondNo8 have a full set of superior figures, unlike their PostScript counterparts. The superior figure glyphs in regular weight only have been copied to `NewG8-sups.pfb` and `NewG8-sups.afm` and provided with a tfm named `NewG8-sups.tfm` that can be used by the superiors package to provide adjustable footnote markers. See superiors-doc.pdf (you can find it in TEXLive by typing `texdoc superiors` in a Terminal window.) The simplest invocation is

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

6 Implementation details

6.1 Small Cap fonts

The small cap fonts were created from the capitals A–Z using FontForge to scale the sizes down uniformly to 67%, then boosting the horizontal and vertical stems up by 130%. The results provided a rough basis for the individual adjustments that had to be made to each glyph. Using FontForge, the stems were

resized appropriately, often requiring a reworking of the shape. The end results are the only possible description of those transformations. Following the creation of those glyphs, appropriate metrics were created using FontForge, the end results of which are provided. The regular weight, upright shape, has been reworked much more than other weights, and looks considerably better, in my opinion. Making a small cap font from scratch takes some real work to get the glyphs, the metrics and the kerning right. In both the regular and bold upright shapes, standard accented glyphs are provided, as well as some special characters and `a_e` and `o_e` ligatures and the glyphs `lslash` and `oslash`.

The small cap macro `\textsc` cooperates with `\textbf` and `\textit`, so you may use, for example:

```
\textsc{Caps and Small Caps}
```

to produce Caps and Small Caps,

```
\textit{\textsc{Caps and Small Caps}}
```

to produce *Caps and Small Caps*,

```
\textbf{\textsc{Caps and Small Caps}}
```

to produce **Caps and Small Caps**, and

```
\textbf{\textit{\textsc{Caps and Small Caps}}}
```

to produce ***Caps and Small Caps***.

6.2 Old style figures

The old style figures were created based on the existing lining figures, reducing the stem lengths of `0` and 1 to lower-case size using FontForge, and lowering the vertical positions of others. The shapes were then modified in FontForge to have more of a traditional oldstyle appearance — the end results show the transformations involved.

7 Matching math packages

Paul Pichaureau's `mathdesign` package has an option `ugm` that sets text to `ugm` and math to his package that matches `ugm`. To use his math package with `garamondx` you write

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

Another possibility is to use the `garamondx` option to `newtxmath`, which uses `garamondx` upper and lower cases italic letters, properly metrized for math, in place of the default Times italics. This requires version 1.06 or higher of the `newtxmath` package.

```
\usepackage{garamondx} % defaults to lining figures, good for math
\usepackage[scaled=.84]{beramono}% good typewriter font
\usepackage[sf,type1]{libertine}%biolinum as sans-serif
\usepackage[garamondx,cmbraces]{newtxmath}
\useosf % changes figure style in garamondx to osf for text, not math
```

Note that the last command, as well as its companion `\useosfI`, may only be used in the preamble, and must not precede `\usepackage{garamondx}`.

8 License

The fonts in this package are derived from the (URW)++ GaramondNo8 fonts which were released under the AFPL, and so the same holds for these fonts. The other support files are subject to the LaTeX Project Public License. See
`http://www.ctan.org/tex-archive/help/Catalogue/licenses.lppl.html`
for the details of that license.

The package and font modifications described above are Copyright Michael Sharpe, msharpe@ucsd.edu, October 31, 2012.

8.1 Font files covered by the 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-Osf-bol.afm
newG8-Osf-bol.pfb
newG8-Osf-bolita.afm
newG8-Osf-bolita.pfb
newG8-Osf-ita.afm
newG8-Osf-ita.pfb
newG8-Osf-reg.afm
newG8-Osf-reg.pfb
NewG8-Reg-SC.afm
NewG8-Reg-SC.pfb
NewG8-Reg.afm
NewG8-Reg.pfb
NewG8-sups.afm
NewG8-sups.pfb
```