# Unslanted Italic Computer Modern Fonts Based on Eric Gill's Ideas

Boris Veytsman*

2010/07/28, v1.1

**Abstract**

This is a demonstration of the use of virtual fonts for unusual effects. We employ virtual fonts to implement an old idea of Eric Gill. This package was written for the author's talk at TUG'2010.

## Contents

## List of Figures

---

*`borisv@lk.net`, `boris@varphi.com`

# List of Tables

# 1 Introduction

This package is mostly a demonstration of the use of Fontinst [1] for mixing and changing fonts. I wrote it as an appendix for my talk at TUG'2010. Of course, as it happens with many teachers, I myself learned a lot while making this demo.

The demo assumes some knowledge of TEX and the way of writing `dtx` files [2].

Eric Gill said that unslanted italics should use Roman uppercase letters [3]:

> Properly speaking, there is no such thing as an alphabet of italic capitals, and where upright or nearly upright italics are used ordinary upright Roman capitals go perfectly well with them.

Let us create a font with italic lowercase and upright uppercase. Actually we create *two* fonts: one based on "normal", i.e. slanted italics, and one based on unslanted ones (we call them `gishape` and `gushape`). In both cases we start with Computer Modern Italics, and perform all changes using virtual fonts mechanism.

Alan Hoenig in his book [4] gives another example of a font based on Gill's idea.

A couple words about the installation of the system. You need the file `http://ctan.tug.org/install/fonts/psfonts/public/gillcm.tds.zip`/ Unzip this file in `$TEXMF`. Add `+cmg.map` to the configuration files of dvips, pdftex and your dvi previewer. How exactly you do this depends on the TEX distribution you use. For TEXLive you create a file `$TEXMF/updmap.d/50gcm.cfg` with just the line

```
Map gcm.map
```

and then run `texhash` and `updmap`. If you use Debian or Ubuntu, the system-wide `updmap.d` directory is located in `/etc`, and you need to create the file `gcm.list` in `/var/lib/tex-common/fontmap-cfg/` with the line

```
50gcm
```

If you use MikTEX2.6, run

```
initexmf --edit-config-file updmap
```

Add to the config file

```
 Map gcm.map
```

save, exit and run `updmap`.

If you use MikTEX2.5 or earlier, edit `localtexmf\web2c\updmap.cfg`, adding the line

```
Map gcm.map
```

and run `updmap`.

After this run updmap and texhash programs to update the configuration files and file names database.

Refer to your distribution documentation for the further details.

`\gishape`
`\gushape`

To use the fonts in LaTeX just add `\usepackage{gillcm}` to the preamble of your document. The style adds two new commands `\gishape` and `\gushape` that typeset "Gillized" italics and "Gillized" upright italics.

# 2 Implementation

## 2.1 Identification

We start with the declaration who we are. Most `.dtx` files put driver code in a separate driver file `.drv`. We roll this code into the main file, and use the pseudo-guard `<gobble>` for it. This means that the text between `<*gobble>` and `</gobble>` will *not* end in any derived file, but it will be used to typeset this manual.

We also use the following guards: `<style>` for LaTeX style, `<drv>` for Fontinst driver `drv.tex`, `<map>` for Fontinst map generator `map.tex` and `<mtx>` for the font metrics file `delupper.mtx`. More about these files below. Note that `mtx` files do not allow LaTeX command `\ProvidesFile`, so we do not put it there.

```
⟨style⟩\NeedsTeXFormat{LaTeX2e}
⟨*gobble⟩
\ProvidesFile{gillcm.dtx}
⟨/gobble⟩
⟨style⟩\ProvidesClass{gillcm}
⟨drv⟩\ProvidesFile{drv.tex}
⟨map⟩\ProvidesFile{map.tex}
⟨*!mtx⟩
[2010/07/28 v1.1 Computer Modern Italics with Gill extensions]
⟨/!mtx⟩
```

And the driver code:

```
⟨*gobble⟩
\documentclass{ltxdoc}
\usepackage{booktabs,amsmath}
\usepackage{url}
\usepackage[breaklinks,colorlinks,linkcolor=black,citecolor=black,
            pagecolor=black,urlcolor=black,hyperindex=false]{hyperref}
\PageIndex
\CodelineIndex
\RecordChanges
\EnableCrossrefs
\begin{document}
  \DocInput{gillcm.dtx}
\end{document}
⟨/gobble⟩
```

```
PFB =\
      cmti10.pfb\
      cmr10.pfb \
      cmbxti10.pfb \
      cmbx10.pfb

AFM = ${PFB:%.pfb=%.afm}

%.pfb:
      cp ‘kpsewhich $@‘ .

%.afm:  %.pfb
      pf2afm $<
```

Figure 1: Creating AFM Files Using `Makefile`

Table 1: Computer Modern Font Files

| File | Font |
|---|---|
| `cmti10.pfb` | Computer Modern Italic |
| `cmr10.pfb` | Computer Modern Roman |
| `cmbxti10.pfb` | Computer Modern Bold Italic |
| `cmbx10.pfb` | Computer Modern Bold |

## 2.2 Preparation of Computer Modern Fonts

We deal with Type 1 fonts, so we need Computer Modern Roman and Italic in the PFB formats. They are provided, for example, by cm-super package. We need the files listed in Table 1. We copy them into the working directory. Since Fontinst prefers ASCII files, we extract metric information from them using `pf2afm` utility.

The famous `make` program allows us to write down these operations succinctly as shown on Figure 1. Here the first paragraph lists our PFB files, the second one says that each PFB file corresponds to one AFM file, the third paragraph tells `make` to copy the PFB file from wherever they are (since `kpsewhich`) to the current directory, and the last paragraph instructs `make` to apply `pf2afm` to each PFB file. Of course if you do not want to use `make`, you can perform all these operations manually.

## 2.3 Fontinst Driver

In this section we construct the file `drv.tex` which does most of the work.

First, we read the Fontinst program `fontinst.sty`:

```
⟨*drv⟩
\input fontinst.sty
```

We want to “unslant” the fonts. This can be imagined as negative slanting. The characteristic angle of Computer Modern Italics is 167/1000 (technically, this is the tangent of the angle, so the angle itself is $\tan^{-1}(0.167) = 9^\circ 28' 51''$), so we use $-167/1000$ as our characteristic angle:

```
\setint{slant}{-167}
```

Now it is time to uncover a white lie in Table 1: the “bold” fonts listed there are actually bold extended fonts: they are slightly widened in the horizontal direction. By tradition we call them nevertheless bold. The next line facilitates this white lie: it instructs Fontinst to silently substitute bold extended fonts whenever a user request bold fonts.

```
\substitutesilent{bx}{b}
```

Fontinst is designed to create virtual font files for TEX. However, it also can be used to automatically create “font maps”: special files read by `dvips` or `pdftex` to map PFB files into the virtual fonts. To do this we need to *record transformations* of Type 1 fonts for the map creation program. Fortunately, Fontinst can do this for us too. The following command starts this recording:

```
\recordtransforms{rec.tex}
```

The command `\endrecordtransforms` below dumps these recordings into the file `rec.tex`. The resulting file is shown on Figure 2.

Now we can create the fonts to play with. We will follow Karl Berry’s scheme [5] for naming them. The font family will be called `cmg` (Computer Modern Gillized). The next letters are `m` for medium, `mi` for medium italic, `miu` for medium italic unslanted, and `b`, `bi`, `biu` for the corresponding bold variants. The last symbols correspond to font encoding.

We start with the fonts in the `8r` (TEX Base 1) encoding.

First, let us create a medium Roman font. We take Computer Modern Roman file `cmr10.afm` and reencode it. The main command is `\trnasformfont`. Its first argument (`cmgm8r`) is the resulting font (remember, the family is `cmg`, the weight is `m` and the encoding is `8r`). Its second argument is the source. This argument says that we take `cmr10.afm` (`\fromafm` command) and reencode it to the required encoding (`\reencodefont`):

```
\transformfont{cmgm8r}{\reencodefont{8r}{\fromafm{cmr10}}}
```

In the same way we create medium italic font `cmgmi8r` from the file `cmti10.afm`:

```
\transformfont{cmgmi8r}{\reencodefont{8r}{\fromafm{cmti10}}}
```

The italic unslanted font `cmgmiu8r` is different: not only we reencode the source, we also unslant it, or, rather, slant it using the negative angle `\int{slant}` defined above. This is done by the command `\slantfont`:

```
\transformfont{cmgmiu8r}{\slantfont{\int{slant}}%
  \reencodefont{8r}{\fromafm{cmti10}}}
```

In the similar manner we create three bold fonts: Roman, italic and italic unslanted:

```
\transformfont{cmgb8r}{\reencodefont{8r}{\fromafm{cmbx10}}}
```

```
\storemapdata{cmr10}{\fromafm{cmr10}{CMR10}}{}
\storemapdata{cmgm8r}{\frommtx{cmr10}}{\reencodefont{8r}}
\storemapdata{cmti10}{\fromafm{cmti10}{CMTI10}}{}
\storemapdata{cmgmi8r}{\frommtx{cmti10}}{\reencodefont{8r}}
\storemapdata{cmti10}{\fromafm{cmti10}{CMTI10}}{}
\storemapdata{cmgmiu8r}{\frommtx{cmti10}}{\reencodefont{8r}\transformfont{1000}{-167}}
\storemapdata{cmbx10}{\fromafm{cmbx10}{CMBX10}}{}
\storemapdata{cmgb8r}{\frommtx{cmbx10}}{\reencodefont{8r}}
\storemapdata{cmbxti10}{\fromafm{cmbxti10}{CMBXTI10}}{}
\storemapdata{cmgbi8r}{\frommtx{cmbxti10}}{\reencodefont{8r}}
\storemapdata{cmbxti10}{\fromafm{cmbxti10}{CMBXTI10}}{}
\storemapdata{cmgbiu8r}{\frommtx{cmbxti10}}{\reencodefont{8r}\transformfont{1000}{-167}}
\makemapentry{cmgmi8r}
\makemapentry{cmgm8r}
\makemapentry{cmgbi8r}
\makemapentry{cmgb8r}
\makemapentry{cmgmiu8r}
\makemapentry{cmgm8r}
\makemapentry{cmgbiu8r}
\makemapentry{cmgb8r}
\makemapentry{cmgmi8r}
\makemapentry{cmgm8r}
\makemapentry{cmgbi8r}
\makemapentry{cmgb8r}
\makemapentry{cmgmiu8r}
\makemapentry{cmgm8r}
\makemapentry{cmgbiu8r}
\makemapentry{cmgb8r}
\makemapentry{cmgmi8r}
\makemapentry{cmgm8r}
\makemapentry{cmgbi8r}
\makemapentry{cmgb8r}
\makemapentry{cmgmiu8r}
\makemapentry{cmgm8r}
\makemapentry{cmgbiu8r}
\makemapentry{cmgb8r}
```

Figure 2: File `rec.tex` (created automatically)

```
\transformfont{cmgbi8r}{\reencodefont{8r}{\fromafm{cmbxti10}}}
\transformfont{cmgbiu8r}{\slantfont{\int{slant}}%
  \reencodefont{8r}{\fromafm{cmbxti10}}}
```

Fonts in TeX Base 1 encoding are useful for manipulation, but they are not directly usable in TeX or LaTeX documents. In this package we create fonts in the traditional TeX encoding (`OT1`) and Cork encoding (`T1`). Fontinst can create for us both virtual font files and LaTeX font description files (`.fd`). This is (rather unfortunately) called font installation. All font installation commands must be bracketed between `\installfonts` and `\endinstallfonts` directives. so we start with the first one:

```
\installfonts
```

Then we declare the family and encoding we are going to install. Let us start with `OT1`. Do not worry about the third (empty) argument of the `\installfamily` command.

```
\installfamily{OT1}{cmg}{}
```

Now the real work begins. Let us construct "Gillized" Computer Modern Italic font. According to our scheme it will be called `cmgmi7t`. This name is the first argument of `\installfont` command.

The second argument is the set of MTX files; one can view it as instructions to Fontinst describing *how* to construct the file. We start from the file `cmgmi8r`, created on the previous step. It is just Computer Modern Italic. The next instruction is `delupper`. It refers to the file `delupper.mtx`, which we create below (Section 2.5). For now it is enough to say that this file deletes from the font all uppercase letters. If we would stop here, the resulting font would have empty spaces instead of letters A, B, … However, we do not stop and add another instruction, `cmgm8r`. Let us recall that `cmgm8r` is Computer Modern Roman. Therefore Fontinst will take the characters from this font and add it to the font created so far. Which characters will be added? The lowercase characters a, b, … are already filled up (Fontinst does not fill the slots already taken), so only empty spaces will be taken from the Roman font—i.e. the spaces that our `delupper.mtx` emptied. We get a combined font! The last instruction, `latin` tells Fontinst how to construct the characters that are absent from the fonts (like the euro sign). It is always prudent to include it as the last resort.

The next argument of the `\installfont` command is the encoding of the result (actually, it can include a list of the so called ETX files).

The last four arguments are used to construct LaTeX font definition files. They say that LaTeX should consider the resulting file to have encoding `OT1`, belong to the family `cmg`, have medium weight (`m`) and "Gill Italic" shape (`gi`). Again, do not mind the last (empty) argument.

So the command looks like the following:

```
\installfont{cmgmi7t}{cmgmi8r,delupper,cmgm8r,latin}{ot1}{OT1}{cmg}{m}{gi}{}
```

We repeat this with bold weight:

```
\installfont{cmgbi7t}{cmgbi8r,delupper,cmgb8r,latin}{ot1}{OT1}{cmg}{bx}{gi}{}
```

Unslanted italics are made in the same way, using `cmgmiu8r` and `cmgbiu8r` as the source:

```
\installfont{cmgmiu7t}{cmgmiu8r,delupper,cmgm8r,latin}{ot1}{OT1}{cmg}{m}{gu}{}
\installfont{cmgbiu7t}{cmgbiu8r,delupper,cmgb8r,latin}{ot1}{OT1}{cmg}{bx}{gu}{}
```

We repeat this with `T1` encoding. Note that for `T1` the "instructions of the last resort" are in the file `newlatin.mtx`:

```
\installfamily{T1}{cmg}{}
\installfont{cmgmi8t}{cmgmi8r,delupper,cmgm8r,newlatin}{t1}{T1}{cmg}{m}{gi}{}
\installfont{cmgbi8t}{cmgbi8r,delupper,cmgb8r,newlatin}{t1}{T1}{cmg}{bx}{gi}{}
\installfont{cmgmiu8t}{cmgmiu8r,delupper,cmgm8r,newlatin}{t1}{T1}{cmg}{m}{gu}{}
\installfont{cmgbiu8t}{cmgbiu8r,delupper,cmgb8r,newlatin}{t1}{T1}{cmg}{bx}{gu}{}
```

Fonts in `T1` encoding should be accompanied by fonts in the "text companion" encoding `TS1`. They include special symbols and accents:

```
\installfamily{TS1}{cmg}{}
\installfont{cmgmi8c}{cmgmi8r,delupper,cmgm8r,textcomp}{ts1}{TS1}{cmg}{m}{gi}{}
\installfont{cmgbi8c}{cmgbi8r,delupper,cmgb8r,textcomp}{ts1}{TS1}{cmg}{bx}{gi}{}
\installfont{cmgmiu8c}{cmgmiu8r,delupper,cmgm8r,textcomp}{ts1}{TS1}{cmg}{m}{gu}{}
\installfont{cmgbiu8c}{cmgbiu8r,delupper,cmgb8r,textcomp}{ts1}{TS1}{cmg}{bx}{gu}{}
```

This finishes our file. Let us just close the open groups:

```
\endinstallfonts
\endrecordtransforms
\bye
⟨/drv⟩
```

## 2.4 Fontmap Generation

The file `map.tex` is used to create font map file. In this section we create it.

First, we read the corresponding Fontinst file:

```
⟨*map⟩
\input finstmsc.sty
```

Next, we tell Fontinst that our Postscript files have the extension `.pfb` and that the resulting map should be called `cmg.map`:

```
\resetstr{PSfontsuffix}{.pfb}
\adddriver{dvips}{cmg.map}
```

Then we read the file `rec.tex` which is generated by the run of `drv.tex` (Figure 2):

```
\input rec.tex
```

And that is all:

```
\donedrivers
\bye
⟨/map⟩
```

## 2.5 Deleting Uppercase Glyphs

Our `\installfont` instructions include deleting all uppercase letters from the italic fonts. This is done by the file `delupper.mtx`, which we describe here.

Everything between `\relax` and `\metrix` is just a comment:

```
⟨*mtx⟩
\relax
Unset all uppercase letters and punctuation
\metrics
```

The file itself uses the command `\unsetglyph`. We unset the uppercase letters, parenthesizes, etc.

```
\unsetglyph{A}
\unsetglyph{B}
\unsetglyph{C}
\unsetglyph{D}
\unsetglyph{E}
\unsetglyph{F}
\unsetglyph{G}
\unsetglyph{H}
\unsetglyph{I}
\unsetglyph{J}
\unsetglyph{K}
\unsetglyph{L}
\unsetglyph{M}
\unsetglyph{N}
\unsetglyph{O}
\unsetglyph{P}
\unsetglyph{Q}
\unsetglyph{R}
\unsetglyph{S}
\unsetglyph{T}
\unsetglyph{U}
\unsetglyph{V}
\unsetglyph{W}
\unsetglyph{X}
\unsetglyph{Y}
\unsetglyph{Z}
\unsetglyph{AE}
\unsetglyph{OE}
\unsetglyph{Oslash}
\unsetglyph{exclam}
\unsetglyph{parenleft}
\unsetglyph{parenright}
\unsetglyph{comma}
\unsetglyph{period}
\unsetglyph{slash}
\unsetglyph{zero}
\unsetglyph{one}
\unsetglyph{two}
```

```
%.map:  drv.tex map.tex $(AFM)
      pdflatex drv.tex
      pdflatex map.tex
      for x in *.pl; do pltotf $$x; done
      for x in *.vpl; do vptovf $$x; done
```

Figure 3: Creating TFM and VF Files using `Makefile`

```
\unsetglyph{three}
\unsetglyph{four}
\unsetglyph{five}
\unsetglyph{six}
\unsetglyph{seven}
\unsetglyph{eight}
\unsetglyph{nine}
\unsetglyph{colon}
\unsetglyph{semicolon}
\unsetglyph{exclamdown}
\unsetglyph{questiondown}
\unsetglyph{question}
\unsetglyph{bracketleft}
\unsetglyph{bracketright}
\endmetrics
⟨/mtx⟩
```

## 2.6 Creation of Fonts

Now we are ready to create fonts. We need to run

```
latex drv.tex
latex map.tex
```

This will create a lot of `.pl` and `.vpl` files. They should be converted into binary format using utilities `pltotf` and `vptovf`. On Figure 3 we show how to do all these tasks using `make`. Of course, we can run all these commands manually.

## 2.7 Style File

To use the fonts in LaTeX let us define commands `\gishape` and `\gushape` that select Gill Italic and Gill Unslanted Italic correspondingly. Note that they should work only in text mode; math mode should use "normal" italics. Here we are:

```
⟨*style⟩
\DeclareRobustCommand\gishape
        {\not@math@alphabet\itshape\mathit
         \fontshape{gi}\fontfamily{cmg}\selectfont}
\DeclareRobustCommand\gushape
```

```
archive: all clean
        $(RM) cmr*.tfm cmt*.tfm cmb*.tfm
        $(RM) cmr*.afm cmt*.afm cmb*.afm
        $(RM) cmr*.vf cmt*.vf cmb*.vf
        mkdir -p fonts/tfm/$(FOUNDRY)/$(PACKAGE)
        cp *.tfm fonts/tfm/$(FOUNDRY)/$(PACKAGE)
        mkdir -p fonts/vf/$(FOUNDRY)/$(PACKAGE)
        cp *.vf fonts/vf/$(FOUNDRY)/$(PACKAGE)
        mkdir -p fonts/map/dvips/$(PACKAGE)
        cp *.map fonts/map/dvips/$(PACKAGE)
        mkdir -p tex/latex/$(PACKAGE)
        cp *.fd *.sty tex/latex/$(PACKAGE)
        mkdir -p doc/latex/$(PACKAGE)
        cp README *.pdf *.tex *.bib *.dtx *.ins doc/latex/$(PACKAGE)
        zip -r $(PACKAGE).tds.zip fonts tex doc
        tar -czvf $(PACKAGE).tgz \
                README $(PACKAGE).tds.zip *.tex *.pdf *.bib *.dtx *.ins
```

Figure 4: Creation of Installation File for CTAN

```
        {\not@math@alphabet\itshape\mathit
         \fontshape{gu}\fontfamily{cmg}\selectfont}
⟨/style⟩
```

## 2.8 Creation of Installation Archive

The last step is to put all the files in the structure suitable for the modern TEX installations (the so called TDS structure). We will create the archive `gillcm.tds.zip` with the relevant files. The archive is packed into the `gillcm.tgz` file ready to be shipped to CTAN. Of course, we do not want to ship Computer Modern fonts, so we delete them from the directory first.

The corresponding fragment of `Makefile` is shown on Figure 4.

This finishes our task!

# 3 Acknowledgements

Most of the things I know about virtual fonts I learned from the excellent book [4] and the manual [6]. I am grateful to TEX Users Group and Karl Berry for encouraging my interest in virtual fonts and to my consulting customers for their support of my font projects.

## References

[1] Alan Jeffrey, Rowland McDonnell, and Lars Hellströom. *Fontinst. Font Installation Software for TEX*, December 2004. `http://mirrors.ctan.org/fonts/utilities/fontinst`.

[2] LATEX3 Project. *LATEX 2ε For Class and Package Writers*, 2006. `http://mirrors.ctan.org/macros/latex/doc/clsguide.pdf`.

[3] Eric Gill. *An Essay on Typography. With a new introduction by Cristopher Skelton.* David R. Godine, Boston, 2007.

[4] Alan Hoenig. *TEX Unbound: LATEX and TEX Strategies for Fonts, Graphics, and More.* Oxford University Press, USA, 1998.

[5] Karl Berry. *Fontname. Filenames For TEX Fonts*, September 2005. `http://mirrors.ctan.org/info/fontname`.

[6] Philipp Lehman. *The Font Installation Guide*, December 2004. `http://mirrors.ctan.org/info/Type1fonts/fontinstallationguide`.

# Change History

v1.0
  General: First fully functional version . . . . . . . . . . . . . . . . . . . . 3

v1.1
  General: Documentation update . . 3

# Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.