cfr-lm

Clea F. Rees*

Version 1.3 — 30th May, 2010

Abstract

cfr-lm is an *experimental* package offering enhanced support for the Latin Modern fonts in TEX/LATEX. A number of features of the Latin Modern fonts are not easily accessible via the default TEX/LATEX support provided in the official distribution. This package aims to provide TEX/LATEX support for a number of these features including various styles of digits, upright italic and oblique small-caps shapes, and alternative weights and widths. It also supports the variable width typewriter, "dunhill" and "quotation" fonts. Version 1.3 supports version 2.004 of the Latin Modern fonts.

Contents

*cfrees <at> imapmail <dot> org

1 Introduction

This document explains how to use the *experimental* cfr-lm package to access advanced features of the Latin Modern fonts not otherwise supported by the official lm distribution. These features include various styles of digits, upright italic and oblique small-caps italic, alternative weights and widths, and Latin Modern Mono Prop (variable width typewriter), Dunhill and Sans Quotation. By default, the LaTeX package provided by cfr-lm.sty uses proportional oldstyle digits and variable width typewriter but this can be changed by passing appropriate options when loading the package. The package also supports using e.g. different styles of digits within a document so it is possible to use proportional oldstyle digits by default, say, but tabular lining digits within a particular table. Finally, a command to access the zeroslash character is provided.

cfr-lm version 1.3 requires version 2.004 of GUST's Latin Modern fonts, including the support package provided for TeX. The fonts and TeX support are included in many TeX distributions or may be obtained from http://www.gust.org.pl/projects/e-foundry/latin-modern or your nearest CTAN mirror.

cfr-lm consists of all files listed in manifest.txt and these files are released under the LaTeX Project Public Licence as explained in the included licensing notices.

Version 1.3 of the package has benefited greatly from feedback provided by Enrico

Gregorio, who essentially rewrote the style file using keyval to show me how I ought to be setting the various options up, and Lars Hellström who demonstrated considerable patience in answering my many questions about using fontinst and some peculiarities of the Latin Modern fonts. I hope the changes in the production of the virtual fonts will improve accent placement in "faked" glyphs (i.e. in the case of characters not included in the EC/T1 font encoding which TEX therefore creates by combining glyphs which are included). The changes involve ignoring all font dimensions given in the AFM files and taking them from the relevant TFM files supplied with Latin Modern instead. The exception to this is the value of `acccapheight` which is set to zero in the TFMs. The current virtual font setup uses fontinst's default value in this case.

These improvements notwithstanding, please note that, unlike the fonts themselves and the official support provided by lm, including `lmodern.sty`, this alternative support package is *experimental*. Although I do not expect the package to raid your fridge or make off with your best computer, it certainly contains bugs and may not work as advertised. If you let me know of problems, I will solve them if I can. If you can correct the problems and send me the fixes, that would be even better.

Although I trust the package enough to use it as my default font package for most purposes, that does not mean you should. You should examine the output especially carefully if you typeset accented characters, especially those which do not occur in English or Welsh text as these are the only languages I typeset more than a word or two of once in a blue moon.

2 Requirements

In addition to the usual suspects (LATEX etc.), the LATEX support provided by `cfr-lm.sty` requires:

- lm: Latin Modern version 2.004[1]
- nfssext-cfr
- xkeyval

The file `clm-test.tex` requires in addition:

- babel
- microtype
- geometry

[1]This package should not be used with any other version of Latin Modern due to likely changes to the font metrics, glyph names etc.

though you can always comment out the relevant lines if you don't have these packages.

In addition to the requirements of `clm-test.tex`, the documentation requires:

- fancyhdr
- lastpage
- fancyref
- array
- longtable
- verbatim
- booktabs
- url

3 Limitations

Unlike the official TEX support for Latin Modern, cfr-lm supports only the EC/T1 and Text Companion (TS1) encodings for text. Also unlike the official support, the EC/T1 support depends entirely on virtual fonts. Whatever the disadvantages of virtual fonts, cfr-lm will inherit them. The support for TS1 and mathematics relies on that provided by the official distribution so should be identical.

4 Installation

As explained in section 2 on the preceding page, the LATEX package cfr-lm requires xkeyval and nfssext-cfr. xkeyval is required for option processing. nfssext-cfr provides additional font selection commands and without it you will get errors complaining that the package cannot be found and you will not be able to use any of the additional font commands described in section 7 on page 11. Both packages are included in TEX Live[2].

Installation varies with TEX distribution so you should consult the documentation which came with your system for details. In most cases, you will need to perform three steps:

1. move or copy the package files to appropriate locations on your system;
2. refresh the TEX database;

[2]This includes MacTEX for OS X users.

3. incorporate the included map file fragments for the different engines your distribution supports.

The following instructions assume you are using. They should not be too difficult to adapt if you are using a different distribution.

4.1 Install the files

The files should be installed in one of two locations: *either* the local system-wide TEX tree *or* your personal tree. If the package is installed system-wide, all users will have access to it. On the other hand, you may need privileges you do not have to do this in which case you must use your personal tree.

For TEX Live, `kpsewhich -var-value TEXMFLOCAL` will return the path to the local tree and `kpsewhich -var-value TEXMFHOME` the path to your personal tree. The package already includes a hierarchy of files to help you install them correctly. Ignoring any symbolic link in the top directory, move or copy the files in `doc`, `fonts` and `tex` into the appropriate locations. If the tree is initially empty, you can simply move or copy the directories in as they are. If the tree already contains other packages, you may need to merge the package hierarchy with the pre-existing one. For example, if you already have a `doc/fonts` directory, move or copy `doc/fonts/cfr-lm` into `doc/fonts/`. If you have a `doc` directory but not a `doc/fonts`, move `doc/fonts` into `doc/`.

4.2 Refresh the database

Again, this depends on your distribution. For TEX Live, `mktexlsr <path to directory>` for the directory you used in the first step should do the trick. Note that you *may* be able to skip this step if you install into your personal tree. Whether this is so depends on the details of your set-up. As a test, move to a directory containing none of the package files and try `kpsewhich cfr-lm.sty`. If the file is found, you don't need to refresh the database; otherwise use `mktexlsr` and then try again.

4.3 Install the map fragments

For TEX Live, there are at least two ways of doing this. The second method varies according to the version of TEX Live and instructions are provided accordingly. Both methods depend on whether you installed into TEXMFLOCAL or TEXMFHOME. If you installed system-wide, the choice is relatively straightforward — it obviously makes sense in that case to update the font maps system-wide as well. If, on the other hand, you installed into your personal tree, the matter is more complex. On the one hand, updating the system-wide maps may create difficulties or confusion for other users because while the map files will list the fonts as available, they will not be able to access them. On the other hand, maintaining personal font map files can produce difficulties and confusions of

its own. Whether it is to be preferred or not is a complex issue and depends on the details of your TEX distribution, local configuration and personal preference. The one clear case is that in which you install into your personal tree because you lack the privileges needed to install system-wide. In that case, you have no choice but to maintain personal font map files or forgo the use of all fonts not provided by your administrator. Other cases are thankfully beyond the scope of this document.

4.3.1 Method 1

If you installed the package system-wide, use the command:

```
updmap-sys --enable Map=clm.map
```

If you installed the package in your personal tree, you *may* prefer to use:

```
updmap --enable Map=clm.map
```

Either way, `updmap` will output a good deal of information after each incantation. This is normal. Just check that it does not end with an error and that it found the new map file.

4.3.2 Method 2: TEX Live 2008 (and probably earlier)

If you installed the package system-wide, use `updmap-sys --edit`.

If you installed into your personal tree, you *may* prefer to use `updmap --edit`.

Either way, a configuration file will be opened which you can edit. Move to the end of the file and add the following line:

```
Map clm.map
```

When you are done, save the file. `updmap` or `updmap-sys` will produce a great deal of output if all is well. Just check that it does not end with an error and that `clm.map` is found.

4.3.3 Method 2: TEX Live 2009 (and possibly later)

If you installed the package system-wide, edit or or create `TEXMFLOCAL/web2c/updmap-local.cfg` and add the following line to the end of the file:

```
Map clm.map
```

Save the file and tell `tlmgr` to merge in your addition using the command:

```
tlmgr generate updmap
```

`tlmgr` will then tell you that you need to ensure the changes are propagated correctly by calling `updmap-sys`. This should produce a great deal of output. Check that it finds the new map file and does not end with an error.

If you installed into your personal tree, you *may* prefer to use `updmap --edit` as described above for TEX Live 2008.

To test your installation and that the package works on your system, latex the file `clm-test.tex`. The console output and/or log should tell you whether any fonts were not found. If you are careful not to overwrite it, you may also compare your output with `clm-test.pdf`.

5 Font setup

As explained above, the fonts use the EC/T1 and Text Companion (TS1) encodings. The provision for the TS1 and mathematics encodings simply calls the support provided by lm. The cfr-lm support simply ensures that access is provided automatically when the T1-encoded virtual fonts it provides are active.

5.1 Font families

The following font families are provided for use in the EC/T1 and Text Companion (TS1) encodings:

LM Names	**Family**	**Digits/figures**	**Notes**
Latin Modern Roman	clm	tabular, lining	similar to lm rm default
	clm2	proportional, lining	
	clmj	tabular, oldstyle	
	clm2j	proportional, oldstyle	cfr-lm rm default
Latin Modern Sans	clms	tabular, lining	similar to lm sf default
	clm2s	proportional, lining	
	clmjs	tabular, oldstyle	
	clm2js	proportional, oldstyle	cfr-lm sf default

LM Names	Family	Digits/figures	Notes
Latin Modern Mono[3]	clmt, clm2t	tabular, lining	similar to lm tt default
	clmjt, clm2jt	tabular, oldstyle	
Latin Modern Mono Prop[4]	clmv	tabular, lining	
	clm2v	proportional, lining	
	clmjv	tabular, oldstyle	
	clm2jv	proportional, oldstyle	cfr-lm tt default
Latin Modern Sans Quotation	clmqs	tabular, lining	
	clm2qs	proportional, lining	
	clmjqs	tabular, oldstyle	
	clm2jqs	proportional, oldstyle	
Latin Modern Roman Dunhill	clmd	tabular, lining	
	clm2d	proportional, lining	
	clmdj	tabular, oldstyle	
	clm2dj	proportional, oldstyle	

5.2 Shapes, weights and widths

family	widths	weights	shapes
clm, clm2, clmj, clm2j	standard	normal	upright, oblique, italic, upright italic, small-caps, oblique small-caps
		bold	upright, oblique, italic
		demi	upright, oblique
clms, clm2s, clmjs, clm2js	standard	normal	upright, oblique
		bold	upright, oblique
	condensed	demi	upright, oblique

[3]The duplication in TEX name here is to avoid TEX complaining if commands to use proportional digits are issued while one of these fonts is active and to ensure that it is possible to switch smoothly to these fonts if another font with proportional digits is active.

[4]Despite the apparent contradiction in their name, this is variable-width typewriter.

family	**widths**	**weights**	**shapes**
clmt, clm2t, clmjt, clm2jt	standard	normal	upright, oblique, italic, small-caps, oblique small-caps
		bold	upright, oblique
		light	upright, oblique
	condensed	light	upright, oblique
clmv, clm2v, clmjv, clm2jv	standard	normal	upright, oblique
		bold	upright, oblique
		light	upright, oblique
clmqs, clm2qs, clmjqs, clm2jqs		normal	upright, oblique
		bold	upright, oblique
clmd, clm2d, clmdj, clm2dj	standard	normal	upright, oblique
		bold	upright, oblique

Where applicable, oblique small-caps are substituted for italic small-caps; italic or oblique for upright italic; oblique for italic; and upright for small-caps. This means that some of the commands described in section 7 on page 11 will fail silently to avoid undue clutter in the log file.

6 The LaTeX package

To load this package, write `\usepackage{cfr-lm}` in your document preamble. By default, the package will define clm2j, clm2js and clm2jv as the default roman/serif, sans and typewriter fonts but you can control the choice by passing options to the package.

The package recognises four keys. Three of these keys take various options which take the value true or false. These control the default style of figures to be used for each of roman/serif, sans and typewriter text, and whether variable or monowidth typewriter will be used by default. The fourth key itself takes a true or false value but has no effect unless `\qtfont` is already defined[5].

[5]This key is designed to control use of LM Sans Quotation in conjunction with prior redefinitions of appropriate environments. Since this is not the sort of redefining a font package should be doing, the option will have absolutely no effect unless you do some prior work to make use of it. In any case, the font can still be accessed directly using the commands explained in section 7 on page 11.

key	affects	option	possible values
rm	default figure style for roman/serif text	oldstyle	true, false
		lining	true, false
		proportional	true, false
		tabular	true, false
sf	default figure style for sans text	oldstyle	true, false
		lining	true, false
		proportional	true, false
		tabular	true, false
tt	default figure style for typewriter text	oldstyle	true, false
		lining	true, false
		proportional	true, false
		tabular	true, false
	default font for typewriter text	monowidth	true, false
		variable	true, false
qt	nothing unless `\qtfont` is defined	—	true, false

The default value in all cases is `true` if an option is given without a value. For example, `rm={oldstyle=true}` is equivalent to `rm={oldstyle}`. Many of the options are provided for ease of use but are essentially equivalent. For example, `proportional=false` is equivalent to `tabular=true`. This means that the following two commands are equivalent:

```
\usepackage[%
rm={lining=true,tabular=false},%
sf={oldstyle,proportional},%
tt={oldstyle=false,proportional=true,monowidth}%
]{cfr-lm}

\usepackage[%
rm={oldstyle=false, proportional=true},%
sf={lining=false,tabular=false},%
tt={lining,proportional,variable=false}%
```

```
]{cfr-lm}
```

Loading the package without options is equivalent to:

```
\usepackage[%
rm={oldstyle=true,proportional=true},%
sf={oldstyle=true,proportional=true},%
tt={oldstyle=true,proportional=true,variable=true},%
qt=false%
]{cfr-lm}
```

That is, by default, oldstyle, proportional figures for roman, sans and typewriter text and variable width typewriter will be selected.

7 Additional font selection commands

cfr=lm loads nfssext-cfr which is an extension of the package nfssext supplied by Philipp Lehman as part of The Font Installation Guide. The file extends the font selection commands to facilitate access to various font features. Both the original and the extension are designed for use with a wide range of fonts. For this reason, only a subset of the additional commands are relevant to any particular font support package. Those relevant to cfr-lm are described below.

I consider my additions to nfssext-cfr to be *highly experimental*. If things don't work as advertised, apart from letting me know about the problem, you may be able to access the features you need by issuing a `\normalfont` and then selecting features from there. This command will return you to the default document text font — typically the relevant serif in regular weight, standard width and upright shape with the style of figures determined by the package and options loaded.

7.1 nfssext-cfr

These commands are available when cfr-lm is loaded. If for some reason you wish to make them available at any other time, use `\usepackage{nfssext-cfr}` in your document preamble.

Note that only combinations supported by the fonts will appear as expected because the commands will only have an effect if the active font offers the relevant variant. For example, trying to switch to a condensed width will have no effect if any of the LM Roman fonts is active. This means that only a subset of combinations are possible. In other cases, one of two things should happen. First, a 'silent' substitution may be made. For example, if you request proportional figures while using monowidth typewriter, tabular figures will be silently substituted. Second, console messages may warn you that the

combination you tried to use isn't available. If you request titling while using monowidth typewriter, a console message will warn you it was unavailable. The file `clm-test.tex` gives an idea of what's possible and also serves as an example illustrating some of the commands provided by cfr-lm and other ways of accessing the fonts..

7.1.1 Widths

width	width command	text command
standard	`\regwidth`	`\textrw{}`
condensed	`\cdwidth`	`\textcd{}`

To switch to an condensed width until further notice, for example, you could use `\cdwidth`. Or use `\texttm{\textlg{\textcd{Hello, world!}}}` to typeset just the text `Hello, world!` in a light-weight condensed monowidth typewriter.

Note that the easiest way to switch to semi-bold condensed sans is to resort to using `\fontseries` directly.

```
\textsf{\fontseries{sbc}\selectfont Semi-bold condensed sans}
```

produces

Semi-bold condensed sans

The problem with using the commands provided by nfssext-cfr is that they are designed, like standard commands such as `\bfseries` and `\scshape`, to change *one* aspect of the font at a time. Issuing `\textsf{\textcd{\textsb{}}}` and `\textsf{\textsb{\textcd{}}}` are equivalent to `\textsf{}` because neither standard-width semi-bold nor condensed normal-weight sans is available. The problem is that each command is processed independently, so both switches fail.

Similar considerations in the case of light condensed monowidth typewriter mean that the *order* in which commands are issued is critical. In this case, a light-weight standard-width font is available, but no normal-weight condensed font is provided. Consequently, `\texttm{\textlg{\textcd{a successful switch}}}` will produce `a successful switch` while `\texttm{\textcd{\textlg{an unsuccessful switch}}}` will result in an `unsuccessful switch` and a warning in the log. In this case, the latter command is equivalent to `\texttm{\textlg{}}` because `\textcd{}` can only succeed *after* `\textlg{}`.

7.1.2 Weights

weight	weight command	text command
light	\lgweight	\textlg{}
semi-bold	\sbweight	\textsb{}

```
\textsb{Semi-bold and \textsl{semi-bold oblique} serif}\\
\texttt{\textlg{Light typewriter}}
```

produces:

Semi-bold and *semi-bold oblique* serif
Light typewriter

7.1.3 Shapes

shape	shape command	text command
oblique small-caps[6]	\sishape	\textsi{}
upright italic	\uishape	\textui{}

```
\textsi{I always avoid a kangaroo.}\\
\textui{Nobody is despised who can manage a crocodile.}
```

produces:

I ALWAYS AVOID A KANGAROO.
Nobody is despised who can manage a crocodile.

if oblique small-caps/upright italic is available for the active font. If it is not, another shape will be substituted.

```
\textsf{\textsi{The bit about the kangaroo was from Lewis Carroll.}}\\
\textbf{\textui{Sylvia snorkeled snappily.}}
```

produces only:

The bits about kangaroos and crocodiles were from Lewis Carroll.
Sylvia snorkeled snappily.

[6] Actually the command switches to *italic* small-caps but since LM does not offer this, oblique small-caps are substituted

where upright sans and bold italic are substituted for italic small-caps sans and bold upright italic since neither is available. Note that the first substitution produces a warning in the log while the second is done "silently".

7.1.4 Figures

Commands are provided to change either one or both aspects of digits' style:

figure style	**style command**	**text command**
lining	`\lstyle`	`\textl{}`
oldstyle	`\ostyle`	`\texto{}`
proportional	`\pstyle`	`\textp{}`
tabular	`\tstyle`	`\textt{}`
proportional, lining	`\plstyle`	`\textpl{}`
proportional, oldstyle	`\postyle`	`\textpo{}`
tabular, lining	`\tlstyle`	`\texttl{}`
tabular, oldstyle	`\tostyle`	`\textto{}`

In this document, proportional, oldstyle figures are used by default for roman/serif, sans and typewriter:

0123456789
0123456789
0123456789

but lining figures are also accessible. For example:

```
\textl{0123456789}\\
\textsf{\textl{0123456789}}\\
\texttt{\textl{0123456789}}
```

produces:

0123456789
0123456789
0123456789

First, note that it is necessary to reissue `\textl{}` after switching to sans or typewriter text. This is because both switching to sans or typewriter and switching to another figure style involves a switch of font family[7].

[7] Compare a switch in width or weight which does not typically involve a change of active font family.

Second, note that the output shows *proportional* lining figures because the command `\textl{}` only changes *one* aspect of the style. Because proportional figures were already active, the command switched to proportional, lining figures. Tabular, lining digits may be accessed in several ways:

```
\textl{\textt{0123456789}}\\
\textt{\textl{0123456789}}\\
\texttl{0123456789}
```

will produce three identical lines of figures:

0123456789
0123456789
0123456789

7.1.5 Typewriter variants

In addition to the package options to specify either LM Mono or LM Mono Prop as default (i.e. either monowidth or variable-width typewriter), it is possible to access the non-default font using the following commands:

typewriter font	style command	text command
variable typewriter	`\tvstyle`	`\texttv{}`
monowidth typewriter	`\tmstyle`	`\texttm{}`

Variable width is default in this document so

```
\texttt{This is variable width typewriter.}\\
\texttm{This is monowidth typewriter} \texttv{except this bit at the end.}
```

produces:

This is variable width typewriter.
This is monowidth typewriter except this bit at the end.

7.1.6 Latin Modern Sans Quotation

This font may be accessed as follows:

font	style command	text command
sans quotation	`\qtstyle`	`\textqt{}`

For example, `\textqt{some text in the font}` will produce some text in the font.

7.1.7 Latin Modern Roman Dunhill

This font may be accessed as follows:

style	style command	text command
titling	`\tistyle`	`\textti{}`

To ensure the command succeeds independently of the currently active font, you may wish to issue `\normalfont` first. For example:

```
\normalfont\textti{Kinky Querulous Rhinos X-Ray Exultant Risque Zebras}\\
\textti{\textsl{Kinky Querulous Rhinos X-Ray Exultant Risque Zebras}}
```

produces:

Kinky Querulous Rhinos X-Ray Exultant Risque Zebras
Kinky Querulous Rhinos X-Ray Exultant Risque Zebras

7.2 zeroslash

cfr-lm provides one additional command. `\zeroslash` will produce the ⌀ character from the current font.