The **nfssext-cfr** package

Clea F. Rees*

2017/03/28

nfssext-cfr is an extension of Philipp Lehman's nfssext. nfssext provides commands which enables one to specify font features not covered by the New Font Selection Scheme of LaTeX 2ε. nfssext-cfr provides additional commands, further extending the facilities offered by NFSS.

Introduction

nfssext-cfr is required by various font support packages I've written. It is being released separately to avoid unnecessary duplication and confusion. At least, I hope it will remove at least one source of unnecessary confusion. I have no reason to think it will avoid any of the others.

The code is somewhat experimental. It works for me. So far. If you discover problems, please let me know. If you know how to fix them, even better.

Caveats, Warnings and Qualifications

The actual effect of any macro depends on any changes made to the defaults for various font features, the current font and, of course, what is available.

For example, `\itdefault` is intended to be the name of italic shape and is used by the redefined `\itshape` supplied by this package. By default, `\itdefault` is `it`. However, if you change that to, say, `sl`, then `\itshape` will use `sl` instead.

Moreover, if the current shape is small-caps, `\itshape` will attempt to merge the default italic shape with the small-caps. That is, it will try to select small-caps italic, if possible, before resorting to plain italic.

The macros operating on family names are almost entirely reliant on font names adhering strictly to the Karl Berry schema. This includes the stipulation that multiple variants be listed in alphabetical order. These macros cannot be used with fonts named in any other way.

Likewise, the macros for series and shapes are unlikely to have their expected effects if fonts are not packaged in ways which both adhere to the NFSS schema and, where relevant, to the specific way that schema is extended here. In particular, note that italic small-caps is assumed to be coded as `si`.

If a macro's attempt to enable or disable a font feature fails, a warning will generally be written to the console, but the code tries hard not to trigger errors. If an attempt triggers an error, that's a bug, so please let me know. If an attempt triggers a warning, please note that there may be no bug at all and, if there is a bug, it is probably not in this package.

To be clear, there certainly are bugs. It is just statistically unlikely that any given warning is caused by one.

Caveat emptor . . .

*ReesC21 <at> cardiff <dot> ac <dot> uk

Macros

The following table includes macros supplied by the original nfssext and additions available with nfssext-cfr.

The third column lists the default letter codes for various font features. As explained above, if the defaults are changed, the macros will try to do something different.

A + indicates that the macro will attempt to merge the addition into the current font's family name, series or shape. For example, if the current font uses oldstyle figures, the `+2` indicates that `\pstyle` will attempt to select a font with figures which are both proportional and oldstyle.

A - indicates that the macro will attempt to subtract from the current font's family name, series or shape. For example, if the current font uses oldstyle figures, the `-2` indicates that `\tstyle` will attempt to select a font with figures which are both tabular and oldstyle.

A comma-separated list indicates consecutive additions and/or subtraction.

If no +- is used, the macro tries to select a font with the given feature without merging. For example `\sistyle` tries to switch to `si` shape regardless of the current font shape.

A – indicates that the macro will try to clear all relevant letter codes from the current font's family name, series or shape. For example, `\regwidth` tries to switch to a series with no letter codes indicating non-standard widths in its name.

Additions, subtractions and clearances operate on font family names, series or shapes, as appropriate. In general, macros with `style` in their names operate on family names; those with `shape` operate on shape codes; and those with width or `weight` operate on series codes.

The letter codes correspond to those specified by the NFSS specification, unless the specification does not include the relevant feature. In the latter case, I tried to choose something sensible i.e. something which made sense to me at the time.

Standard macros redefined:			
\itshape			
\scshape			
\upshape			
Families — Styles:			
\textti	\tistyle	+d	titling/display
\textlt	\ltstyle	+l	light (when separate family)
\textof	\ofstyle	+l	open-face (or outline or blank) style
\textalt	\altstyle	+a	alternative style
\textreg	\regstyle	–	regular style
\emboss	\embossstyle	+e	
\textorn	\ornamentalstyle	+p	intended primarily for decorative initials etc.
\ornament			
\textqt	\qtstyle	+q	quotation style
\textsh	\shstyle	+h	shadowed style
\texttm	\tmstyle	-s,-v,+t	monowidth typewriter
\texttv	\tvstyle	-s,-t,+v	variable width typewriter
\textswash	\swashstyle	+w	an attempt to improve on \textsw
Families — Figures:			
\textln	\lnstyle	–	lining figures (cf. macro below)
\textos	\osstyle	j	oldstyle figures (cf. macro below)
\textin	\instyle	o	inferior figures
\textsu	\sustyle	1	superior figures
\textl	\lstyle	-j	lining figures (cf. command above)

\texto	\ostyle	+j	oldstyle figures (cf. original \osstyle above)
\textp	\pstyle	+2	proportional figures
\textt	\tstyle	-2	tabular figures
\textpl	\plstyle	-j,+2	proportional lining figures
\textpo	\postyle	+2j	proportional oldstyle figures
\texttl	\tlstyle	-j,-2	tabular lining figures
\textto	\tostyle	+j,-2	tabular oldstyle figures
Shapes:			
\scolshape		scol	
\textol	\olshape	ol	outline
\textsi	\sishape	si	italic small-caps
\textu	\ushape	u	
\textscu	\scushape	su	
\textui	\uishape	ui	upright italic
\textri	\rishape	ri	reverse italic
\textdf	\dfshape	n	default shape
\textsw	\swshape	it	swash shape (cf. \swstyle above)
	\swstyle		
Series — Widths:			
\textnw	\nwwidth	+c	
\textcd	\cdwidth	+c	
\textec	\ecwidth	+ec	
\textuc	\ucwidth	+uc	
\textet	\etwidth	+x	
\textep	\epwidth	+x	
\textex	\exwidth	+ex	
\textux	\uxwidth	+ux	
\textrw	\regwidth	–	
Series — Weights:			
\textmb	\mbweight	+mb	
\textdb	\dbweight	+db	
\textsb	\sbweight	+sb	
\texteb	\ebweight	+eb	
\textub	\ubweight	+ub	
\textlg	\lgweight	+l	
\textel	\elweight	+el	
\textul	\ulweight	+ul	

History

2017-03-28

Attempt to modify \tmstyle and \tvstyle to work as advertised. Extend documentation somewhat.

2010

The 2010 update includes an attempt to improve the behaviour of \ofstyle, and to add support for microtype. I didn't publish this at the time because I wanted to test it first. I have just discovered

that I am still using a local copy. Insofar as one person can test something, I figure that 5 years ought to be enough to pick up the most obvious problems. However, your kilometres may, as always, vary.

There should be no changes for the end user except that in certain cases it is possible that line-breaks may be altered if microtype is in use due to the enhanced support included for variant font families.