

nfssex-cfr

Clea F. Rees*

v1.0 (SVN Rev: 10366) 2024/09/18

Abstract

`nfssex-cfr` is an extension and modification of Philipp Lehman's `nfssex` which provides extended font selection commands modelled on those provided by $\text{\LaTeX 2}_{\epsilon}$. Given an appropriate font configuration, `nfssex-cfr` enables users to change the weight, width, shape and style of font as easily as they can select bold, italic or typewriter. For instance, the package makes it trivial to use proportional, hanging figures in the body of the text, proportional, lining figures in captions and headers and tabular, lining figures in tables. An extensive choice of commands are provided to access a wide variety of weights, widths, shapes and styles from the more common (e.g. semi-bold or condensed) to the less common (e.g. 'outline' and right or upright italic). Comprehensive support is provided for 'swash' and 'alternate' styles. These are implemented as families rather than shapes to support fonts which offer multiple swash shapes (e.g. small-caps, italic and upright) or alternate styles. These may be used to provide effective access to fancy ligatures, end-of-word swashes etc. without sacrificing the range of characters provided by `T1`.

The package is not primarily intended for direct use by end-users, but is designed rather to facilitate the creation of more sophisticated font support packages. End-users may nonetheless find the package useful, subject to the constraints explained in this document. Moreover, end-users may wish to pass options to the package on newer kernels, if loading font support packages which have not been updated for changes to font selection¹.

Contents

1	Introduction	2
2	Macros	3
3	Newer \LaTeX Kernels	8
3.1	Required Changes to Font Support Files	9
4	Older \LaTeX Kernels	10

*Bug tracker: codeberg.org/cfr/nfssex/issues | Code: codeberg.org/cfr/nfssex | Mirror: github.com/cfr42/nfssex

¹'Kernel' refers to the \LaTeX kernel in this context and should not be confused with your system kernel.

5	Bugs, Non-Bugs & Debugging	11
6	Implementation	12
6.1	Main package file	12
6.2	NNFSS	16
6.3	NFSS	36

List of Tables

1	Standard (kernel) macros (re)defined	4
2	Standard (kernel) font change rules redefined	4
3	Family switches: general	5
4	Family switches: figures	6
5	Shape switches	6
6	Series switches: widths	7
7	Series switches: weights	7

1 Introduction

The package was originally a fairly simple extension of Philipp Lehman’s `nfssect`. `nfssect` provides commands which enable one to specify font features not covered by the New Font Selection Scheme (NFSS). The package developed according to the needs of particular fonts I configured for L^AT_EX and, in a few cases, my dissatisfaction with the original commands.

In adapting the package for the (New) New Font Selection Scheme² (NNFSS), I have tried to balance (i) backwards compatibility³ for users loading updated font support packages, (ii) backwards compatibility for users loading packages which haven’t been updated and (iii) compatibility with the new features of NNFSS. I have also tried to account for the common case in which documents use combinations of fonts from different packages, each of which may or may not load `nfssect-cfr` and may or may not have been updated for NNFSS. This has inevitably required some compromises and there are certainly places where I would do things differently if starting from scratch.

While I don’t recommend installing this version of `nfssect-cfr` on an older system, the package should continue to work more-or-less as it always did on older kernels. To achieve this, the package is split into a main file, `nfssect-cfr.sty`, which provides common code and figures out whether to load code for NFSS (`nfssect-cfr-nfss.sty`) or NNFSS (`nfssect-sty-nnfss.sty`).

²Officially, there is no such designation, but I have to call it something. Throughout this document and packages which depend upon it, I use this term to refer to the font selection features introduced into L^AT_EX in 2020.

³Note that 100% backwards compatibility cannot be implemented on current L^AT_EX kernels.

`nfssex-cfr-nfss.sty` is essentially what was `nfssex-cfr.sty` minus the code retained in the latter.

Unfortunately, it is impossible to ensure 100% backwards compatibility with recent L^AT_EX kernels. By default, `nfssex-cfr` tries to interfere as little as possible with the kernel, even at the cost of backwards compatibility. If `compat` is enabled, however, the package does its best to enable backwards compatible behaviour, at the cost of the new functionality provided by the kernel.

`nfssex-cfr` does not eschew interference with the kernel at all costs. Even without `compat` it patches or replaces some kernel code because some things just don't work sensibly⁴.

All font-support code should be updated to use `compat=false` when loading `nfssex-cfr`. The changes in NNFSS **require** changes to code based on Philipp Lehman's Font Installation Guide.

2 Macros

Tables 1 to 7 include macros supplied by the original `nfssex` and additions available with `nfssex-cfr`. Macros in tables 5 to 7 should work with any font definition files which more-or-less adhere to NFSS/NNFSS. This should, theoretically, be all font packages but, in practice, things are rarely so simple. Macros in tables 3 and 4 will work only with fonts named strictly according to the Berry naming scheme.

In tables 3 to 7, the third column lists the default letter codes for various font features. 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

⁴This is true in two main places. The first is the kernel's initialisation of series at the start of the document. This overwrites the default **bf** series according to the font family name rather than the font name. This means that virtual fonts which depend on Computer or Latin Modern are not handled correctly and, because this code is delayed, the problem cannot be corrected by setting things up appropriately earlier.

The second is the implementation of 'swash' which is by far the most problematic of the changes and one of the most difficult to navigate. It isn't clear to me how seriously the kernel's definition is intended to be, but I have chosen to overwrite the kernel code here.

Table 1: Standard (kernel) macros (re)defined

<code>\swshape</code> ^b	redefined on new kernels ^c defined on old kernels
<code>\itshape</code>	old kernels only
<code>\scshape</code>	old kernels only
<code>\upshape</code>	old kernels only

^a Defined only by newer kernels.^b See tables 3 and 5 and text.^c Definition depends on kernel, `force` and `compat`.

Table 2: Standard (kernel) font change rules redefined

Shape		
Current	Requested	Applied when?
<code>it</code>	<code>sc</code>	<code>compat</code> & NNFSS only
<code>sl</code>	<code>sc</code>	<code>compat</code> & NNFSS only
<code>sc</code>	<code>it</code>	<code>compat</code> & NNFSS only
<code>sc</code>	<code>sl</code>	<code>compat</code> & NNFSS only
<code>scsl</code>	<code>it</code>	<code>compat</code> & NNFSS only

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 codes⁵; 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. These choices are not always those specified by the NNFSS specification, since sense and sensibility are sometimes in the eye of the encoder.

One further macro is available, though it has no effect on older kernels.

`\nfssectset` $\{\langle\text{key-value list}\rangle\}$

Package options (see section 3) may also be specified after loading either in the preamble (`compat` and `force`) or at any time (`debug`). This enables users to set options after some other package loads `nfssect-cfr` and allows additional information to be printed to the console and/or logged on local basis.

⁵But **not** `\swshape`!

Table 3: Family switches: general

Macros		Family Code	Style
Text Command	Switch		
<code>\textti</code>	<code>\tistyle</code>	+d	titling/display
<code>\textlt^a</code>	<code>\ltstyle^b</code>	+l	light if separate family
<code>\textof</code>	<code>\ofstyle</code>	+l	open-face (or outline or blank) style
<code>\textalt</code>	<code>\altstyle</code>	+a	alternative style
<code>\textreg</code>	<code>\regstyle</code>	-	regular style
<code>\emboss</code>	<code>\embossstyle</code>	+e	‘embossed’ style
<code>\textorn</code>	<code>\ornamentalstyle</code>	+p	decorative initials etc.
<code>\ornament</code>			
<code>\textqt</code>	<code>\qtstyle</code>	+q	quotation style
<code>\textsh</code>	<code>\shstyle</code>	+h	shadowed style
<code>\texttm</code>	<code>\tmstyle</code>	-s,-v,+t	monowidth typewriter
<code>\texttv</code>	<code>\tvstyle</code>	-s,-t,+v	variable width type-writer
<code>\textswash</code>	<code>\swashstyle^c</code>	+w	swash
<code>\textsw^d</code>	<code>\swshape^d</code>		‘find a route to swash’

^a Cf. `\textlg` in table 7.

^b Cf. `\lgweight` in table 7.

^c Cf. `\swstyle` in table 5.

^d Effect is kernel and option dependent, but potentially changes family and/or shape. ‘Tries to find a route to swash.’ See text for an explanation of what, why and when. See section 6 for details of how.

Table 4: Family switches: figures

Macros		Family Code	Style of Figures
Text Command	Switch		
<code>\textln^a</code>	<code>\lnstyle^a</code>	–	lining (cf. <code>\lstyle</code> below)
<code>\textos^a</code>	<code>\osstyle^a</code>	j	oldstyle (cf. <code>\ostyle</code> below)
<code>\textinf</code>	<code>\infstyle</code>	0	inferior
	<code>\instyle</code>		
<code>\textin^b</code>			if hyperref is not loaded
<code>\textsu</code>	<code>\sustyle</code>	1	superior
<code>\textl^c</code>	<code>\lstyle^c</code>	–j	lining (cf. <code>\lnstyle</code> above)
<code>\texto^c</code>	<code>\ostyle^c</code>	+j	oldstyle (cf. <code>\osstyle</code> above)
<code>\textp^c</code>	<code>\pstyle^c</code>	+2	proportional
<code>\textt^c</code>	<code>\tstyle^c</code>	–2	tabular
<code>\textpl^d</code>	<code>\plstyle^d</code>	–j,+2	proportional lining
<code>\textpo^d</code>	<code>\postyle^d</code>	+2j	proportional oldstyle
<code>\texttl^d</code>	<code>\tlstyle^d</code>	–j,–2	tabular lining
<code>\textto^d</code>	<code>\tostyle^d</code>	+j,–2	tabular oldstyle

^a This macro is the original `nfssect` command. The result is independent of the current style of figures.

^b Deprecated.

^c This macro changes precisely one aspect of the current figure style. That is, the result depends on the current style of figures.

^d This macro ensures a specific fully-specified figure style.

Table 5: Shape switches

Macros		Shape Code	Shape
Text Command	Switch		
–	<code>\scolshape</code>	scol	outline small-caps
<code>\textol</code>	<code>\olshape</code>	ol	outline
<code>\textsi</code>	<code>\sishape</code>	si	italic small-caps
<code>\textu</code>	<code>\ushape</code>	u	??
<code>\textscu</code>	<code>\scushape</code>	su	??
<code>\textui</code>	<code>\uishape</code>	ui	upright italic
<code>\textri</code>	<code>\rishape</code>	ri	reverse italic
<code>\textdf</code>	<code>\dfshape</code>	n	default shape
–	<code>\swstyle^b</code>	+w,it	swash family <i>and</i> shape
<code>\textsw^a</code>	<code>\swshape^c</code>		‘find a route to swash’

^a Cf. `\textswash` in table 3.

^b Cf. `\swashstyle` in table 3.

^c Definition is kernel and option dependent, but probably doesn’t (just) change shape. See table 3 for sketch and text for details.

Table 6: Series switches: widths

Macros		Switch	Series Code	Width
Text	Command			
	<code>\textnw</code>	<code>\nwwidth</code>	<code>+c</code>	narrow
	<code>\textcd</code>	<code>\cdwidth</code>	<code>+c</code>	compact
	<code>\textec</code>	<code>\ecwidth</code>	<code>+ec</code>	extra compact
	<code>\textuc</code>	<code>\ucwidth</code>	<code>+uc</code>	ultra compact
–		<code>\mdwidth</code>	<code>+?m</code>	medium
	<code>\textet</code>	<code>\etwidth</code>	<code>+x</code>	extended
	<code>\textep</code>	<code>\epwidth</code>	<code>+x</code>	expanded
	<code>\textex</code>	<code>\exwidth</code>	<code>+ex</code>	extra expanded
	<code>\textux</code>	<code>\uxwidth</code>	<code>+ux</code>	ultra expanded
	<code>\textrw</code>	<code>\regwidth</code>	–	regular

Table 7: Series switches: weights

Macros		Switch	Series Code	Weight
Text	Command			
–		<code>\mdweight</code>	<code>+m?</code>	medium
	<code>\textmb</code>	<code>\mbweight</code>	<code>+mb</code>	medium-bold
	<code>\textdb</code>	<code>\dbweight</code>	<code>+db</code>	demi-bold
	<code>\textsb</code>	<code>\sbweight</code>	<code>+sb</code>	semi-bold
	<code>\textbd</code>	<code>\bdweight</code>	<code>+b</code>	bold
		<code>\bfweight</code>		
	<code>\texteb</code>	<code>\ebweight</code>	<code>+eb</code>	extra-bold
	<code>\textub</code>	<code>\ubweight</code>	<code>+ub</code>	ultra-bold
	<code>\textlg^a</code>	<code>\lgweight^b</code>	<code>+l</code>	light when weight
	<code>\textel</code>	<code>\elweight</code>	<code>+el</code>	extra-light
	<code>\textul</code>	<code>\ulweight</code>	<code>+ul</code>	ultra-light

^a Cf. `\textlt` in table 3.^b Cf. `\ltstyle` in table 3.

3 Newer L^AT_EX Kernels

The package tests for the presence of `\init@series@setup`. If this exists, it loads a newer version of the package. `nfssect-cfr` supports three options, but these are only effective if the newer code (for NFSS) is loaded. All three are booleans, initially false and default to true if used without specifying a value⁶. The third option is described in section 5.

`force (opt.) = true|false`

Default: `true`

Initial: `false`

Scope: preamble

You can force the old code to be loaded using the package option `force` or `force=true`. Note, however, that the old version will not work as advertised on newer kernels because L^AT_EX will overwrite some of the package's definitions at the end of the preamble.

`compat (opt.) = true|false`

Default: `true`

Initial: `false`

Scope: preamble

In contrast, `compat` or `compat=true` will activate code which tries to partially replicate the original `nfssect-cfr`'s behaviour. This is far from unproblematic. In particular, it will partly break features of the current NFSS for other fonts.

If your document relies exclusively on text fonts supported by this package and none of the support for those fonts has been updated, compilation should produce a more satisfactory result than otherwise. If, however, your document relies partly on text fonts not supported by this package or the support for those fonts has been updated in the 'wrong' way, compilation may produce a less satisfactory result. There is no general rule here: whether the option helps or hinders things depends entirely on the fonts, the support for those fonts and the specific contents of your document.

The main areas known to be problematic are

1. italic small-capitals (but oblique small-capitals should be mostly unaffected);
2. swash;
3. transitions between small-caps, italic, oblique, italic small-caps, oblique small-caps, upright italic, right italic and upright;
4. any transition involving swash where shape is involved;

⁶Default' and 'initial' follow the usage in `l3interface.pdf`, `l3keys2e` and `clsguide.pdf`. If you are familiar with `pgfkeys`, the terms have the same meaning there. Basically, the 'initial' value is what you get if you don't specify the option at all when loading the package, while the 'default' is what you get if you specify the option without specifying a value.

'Scope' is used in the standard sense applicable to L^AT_EX 2_ε class and package options. That is, it indicates whether the option may be used only when loading the package, at any point in the preamble or also in the document.

5. medium weight fonts where width is non-standard e.g. medium condensed, medium ultra condensed etc.;
6. medium bold weight in any context.

1–4 can be worked around at the document level, with some inconvenience. Subject to the caveats above, the `compat` option may avoid at least some of these inconveniences.

5 and 6 cannot be worked around at the document level. Nor does `nfssex-cfr` make any attempt to mitigate these two issues. Doing so would involve too much interference with current NNFSS. This means that 5 and 6 can be addressed only in the support files for the fonts affected. Neither `compat` nor `force` makes any attempt to change this.

Maximum backwards compatibility requires changes to the font support files *and* `compat=false`, but some documents may still require (hopefully minor) changes.

3.1 Required Changes to Font Support Files

In all cases, additional changes to font substitution rules may be needed to prevent multiple substitutions by the same font, since these seem to cause problems.

italic small-caps The problem here is that `nfssex` encoded italic small-caps as shape `si`, whereas the kernel has plumped for `scit`. It does support `scsl` (although it does not distinguish oblique from italic), but not `si`. This issue can be more-or-less dealt with by support files for fonts, but some issues arise at the document level concerning transitions (below).

Ideally, `scit` should be substituted wherever font definition files specify the shape `si`. `scit` should then be defined as a (silent) substitution for `si`. However, it *should* be sufficient to define `si` as a substitute for `scit`.

Fonts which provide oblique small-caps, but not italic, should specify `scsl` as a (silent) substitution for `scit` and `scit` for `si` (or *vice-versa*). The kernel supports `scsl` out-of-the-box, together with the substitution for `scit`, but the changes should make support for `\textsi` and `\sishape` more robust.

transitions No additional changes are recommended to support files which load `nfssex-cfr`. Provided `fd` files are updated as explained above, no further adjustments should be required to enable correct font selection during transitions involving italic, oblique, small-caps, upright italic, reverse italic etc.

swash No changes are recommended for swash to font definition files for packages which load `nfssex-cfr`. If loading `nfssex-cfr`, the recommendation is to **ignore** the kernel's implementation because it cannot be made to work correctly with any family which provides swash for multiple shapes e.g. both upright and italic or small-caps and upper/lower case⁷.

⁷The issue here is that the kernel considers swash to be a *shape*, whereas `nfssex` only *called* it a shape. The underlying code treated it as a *style* requiring a change of font *family*. The shape

swshape No changes for swash are recommended for `sty` files in packages which load `nfssect-cfr`.

`nfssect-cfr-nfss` contains the original `nfssect` definition of `\swshape` and `\textsw`. This is used on newer kernels only if `force` is used, in which case the code is largely broken.

`nfssect-cfr-nfss` contains both the original definition and a replacement. The former is used only if `compat` is selected; otherwise, the latter is used on kernels supporting NFSS. The new definition tries to figure out which of the three possible implementations, if any, to use and behaves accordingly. **The kernel definition is overwritten regardless.** `compat` determines only *what* overwrites the kernel's.

medium Any line of a font definition file which codes a series of two or more letters including `m` must be changed to delete the `m`. For example, `{mc} → {c}`, `{muc} → {uc}` etc. **It is NOT sufficient to substitute such series using rules.** The changes **must** be made in the primary definitions of the font families.

mb I've chosen to make `\mbweight` an alias for `\sbweight`. Hopefully no font family supports both. Provided that's not the case, `mb` should be changed to `\sb` in all affected font definition files. **It is NOT sufficient to substitute `mb` for `sb` using a rule.** The change **must** be made in the definition of the family.

4 Older L^AT_EX Kernels

This is the code base `nfssect` was written for. `nfssect-cfr` extended that code.

To the best of my knowledge, the code used with older L^AT_EX kernels works as expected. This means it is fully compatible with the Font Installation Guide and that things like `\scshape` `\itshape` will produce italic small-caps, as expected. This code is also — again to the best of my knowledge — fully compatible with all features of NFSS with the single exception of code supporting medium weight, condensed width fonts which erroneously uses `mc` rather than `c`.

Italic small-caps is assumed to be coded as `si`. Oblique small-caps is assumed to be coded as `scl`.

If a set of fonts provides a swash *family*, it is assumed the fonts are named in accordance with the Berry scheme. `\textswash`, `\swshape`, `\swashstyle` and/or `\swstyle` can then be used to access this family.

The difference between `\swashstyle` and `\swstyle` is that the former tries to merge any swash family with the current one, whereas the latter does not. So, if

was always (potentially switched to) italic (`it`). `nfssect-cfr` offered a second version of swash, which treats it as a family possibly requiring a change of shape, but the shape is typically italic or upright, as opposed to being specific to swash.

To make things worse, not all fonts *can* be setup in the way the kernel assumes because some fonts provide swash characters in a variety of shapes (upright and italic, for example). Moreover, it is common to encode additional ligatures, for example, as swash, even though this is not accurate, in order to provide *some* mechanism for accessing them within a traditional 8-bit font setup.

a font set provides swash for two widths of font, say, and you've changed widths, `\swashstyle` will try to find a swash character without altering the width, whereas `\swstyle` will first switch to the base font, resetting the width. On the other hand, if the font only provides swash in the standard width, say, and you've changed widths, `\swashstyle` will fail to switch to swash, whereas `\swstyle` will succeed. Packages which include swash families should, therefore, advise users which command(s) to use.

`\textswash` is the text font command for `\swashstyle`. `\textsw` is the text font command for `\swshape`.

`\swshape` first tries `\swstyle` before changing the shape to `\swshapedefault`. By default, this is `\itdefault` because swash families are often coded as italic, but this is obviously font-dependant.

5 Bugs, Non-Bugs & Debugging

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.

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.

On older kernels (NFSS), changes to weight and width should work and most shapes should be supported, but italic small-caps is assumed to be coded as `si` on these kernels, so you may need something like

```
\renewcommand*{\sidefault}{scit}% or scsl or whatever
```

On newer kernels, italic small-caps should be encoded as explained above and `\sishape/\textsi` should work out-of-the-box with packages which use any of `si`, `scit` or `scsl`.

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 package⁸

```
debug (opt.) = true|false
```

Default: `true`

Initial: `false`

Scope: general

You can get a bit more information printed to the console about what's happening using this option on newer kernels. Some of the same information can also be

⁸To be clear, there certainly are bugs. It is just statistically unlikely that any given warning is caused by one.

found in the log.

6 Implementation

You do not need to read the remainder of this document in order to install or use the package.

6.1 Main package file

This used to be the complete package. Now it is mostly responsible for processing options, figuring out which kernel we're on and implementing the small amount of code shared between the implementations for NFSS and NNFSS.

`nfssect-cfr` (*pkg.*)

```

1 \NeedsTeXFormat{LaTeX2e}
2 \RequirePackage{svn-prov}
3 \ProvidesPackageSVN[\filebase.sty]{$Id: nfssect-cfr.dtx 10366 2024-09-18
  14:25:21Z cfrees $}[v1.0 \revinfo{} extensions for NFSS and NNFSS; based
  on 2003/03/14 v1.2 Experimental NFSS Extensions]
4 \DefineFileInfoSVN

```

`\if@nfssectcfr@digonnew`

```

5 \newif\if@nfssectcfr@digonnew
6 \@nfssectcfr@digonnewtrue

```

Copied verbatim, excepting format and modulo package/module name from Joseph Wright's `siunitx.sty` under LPPL

```

7 \@ifundefined{ExplLoaderFileDate}{%
8   \IfFileExists{expl3.sty}{%
9     \RequirePackage{expl3}%
10    }{%
11     \RequirePackage{nfssect-cfr-nfss}%
12     \@nfssectcfr@digonnewfalse
13    }%
14 }{}
15 \if@nfssectcfr@digonnew

```

Almost verbatim from `siunitx.sty`

```

16 \ifl@t@r\ExplLoaderFileDate{2022-02-24}{%
17   }{%
18     \RequirePackage{nfssect-cfr-nfss}%
19     \@nfssectcfr@digonnewfalse
20   }%
21 \fi
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 \if@nfssectcfr@digonnew

```

BEGIN expl pkg option setup

```

24 \newif\ifexfs@debug
25 \ExplSyntaxOn
26 \prop_gput:Nnn \g_msg_module_name_prop { nfssect-cfr } { exfs }
27 \keys_define:nn { exfs }
28 {

```

`compat (opt.)` Compatibility option.

```

\g__exfs_compat_bool
29 compat .bool_gset:N = \g__exfs_compat_bool,
30 compat .default:n = true,
31 compat .initial:n = false,
32 compat .usage:n = preamble,

```

`debug (opt.)` Turns info messages into warnings for testing purposes and possibly provides additional information.

```

33 debug .legacy_if_gset:n = exfs@debug,
34 debug .initial:n = false,
35 debug .default:n = true,

```

`force (opt.)` Force loading of code for NFSS even on newer kernels.

```

\g__exfs_force_bool
36 force .bool_gset:N = \g__exfs_force_bool,
37 force .default:n = true,
38 force .initial:n = false,
39 force .usage:n = preamble,

40 }

```

Joseph Wright: from siunitx.sty ; <https://chat.stackexchange.com/transcript/message/64327823#64327823>

```

41 \providecommand \IfFormatAtLeastTF { \@ifl@t@r \fmtversion }
42 \IfFormatAtLeastTF { 2022-06-01 }
43 {
44   \ProcessKeyOptions [ exfs ]
45 }{
46   \RequirePackage { l3keys2e }
47   \ProcessKeysOptions { exfs }

```

`\ProcessKeyOptions`

```

48 \NewDocumentCommand \ProcessKeyOptions { o }
49 {
50   \IfValueTF { #1 } { \ProcessKeysOptions { #1 } }
51   { \PackageError{nfssect-cfr}{
52     Optional~argument~mandatory~on~kernels~this~old.\MessageBreak
53     Please~specify~the~module~whose~keys~should~be~processed
54   }{
55     \protect\ProcessKeyOptions~only~passes~keys~to~
56     \protect\ProcessKeyOptions\MessageBreak
57     on~older~kernels~for~the~convenience~of~package~authors.~
58     Since~\protect\ProcessKeysOptions\MessageBreak

```

```

59         takes-an~argument,~the~optional~argument~to~
60         \protect\ProcessKeyOptions~is~required
61     }
62 }
63 }

64 }
65 \IfFormatAtLeastTF { 2020-10-01 }{
66 }{
67   \RequirePackage { xparse }

```

\ExpandArgs

```

68   \providecommand \ExpandArgs [1]
69   { \cs_if_exist_use:c { exp_args:N #1 } }

70 }

```

END expl pkg option setup

BEGIN cfr-added: bifurcate

We test for the presence of \init@series@setup in order to determine whether to load code for NNFSS or NFSS. If force is set, we load for NFSS regardless.

```

71 \RequirePackage{etoolbox}
72 \msg_new:nnn { nfssect-cfr } { compat }
73 {
74   You ~ or ~ a ~ font-support ~ package ~ have ~ loaded ~ me ~ ( line ~ \msg_line_number:
75   ) ~ with ~ the ~ compat ~ option. ~
76   This ~ means ~ the ~ package ~ may ~ require ~ updating. ~
77   Update ~ the ~ package ~ if ~ necessary ~ and ~ use ~ compat=false ~ when
78   ~ loading ~ me.
79 }
80 \msg_new:nnn { nfssect-cfr } { force }
81 {
82   You ~ or ~ a ~ font-support ~ package ~ have ~ loaded ~ me ~ ( line ~ \msg_line_number:
83   ) ~ with ~ the ~ force ~ option. ~
84   This ~ is ~ a ~ desperate ~ measure ~ of ~ last ~ resort. ~
85   **Breakage ~ is ~ expected.**
86 }
87 \hook_gput_code:nnn { begindocument/before } { . }
88 {
89   \cs_if_exist:NTF \init@series@setup
90   {
91     \bool_if:NTF \g__exfs_force_bool
92     {
93       \msg_warning:nn { nfssect-cfr } { force }
94       \RequirePackage {nfssect-cfr-nfss}
95     }{
96       \bool_if:NT \g__exfs_compat_bool
97       {
98         \msg_warning:nn { nfssect-cfr } { compat }
99       }
100       \RequirePackage {nfssect-cfr-nnfss}
101     }
102   }

```

```

99   }{% achosiad heb broblemau - diolch byth!
100   \RequirePackage {nfssect-cfr-nfss}
101   }
102 }

```

`__exfs_set:n`

```

103 \cs_new_protected_nopar:Nn \__exfs_set:n
104 {
105   \keys_set:nn { exfs } { #1 }
106 }

```

`\nfssectset` Allow setting of options later in preamble or in document. This is intended to enable debugging to be toggled locally.

```

107 \cs_set_eq:NN \nfssectset \__exfs_set:n
108 \ExplSyntaxOff

```

END added

`\DeclareTextOrnament` These are unmodified from `nfssect`. I'm not aware of any [CTAN](#) packages using `\ornament` these and they have not been tested for compatibility with NNFSS, though I can't see any patently obvious problems.

```

109 \newcommand*{\DeclareTextOrnament}[7]{%
110   \expandafter\def\csname#1@orn\@roman#2\endcsname{#3/#4/#5/#6/#7}}
111 \begingroup
112   \catcode'\/=12
113   \gdef\exfs@split@orndef#1/#2/#3/#4/#5\@nil{%
114     \def\f@encoding{#1}%
115     \def\f@family{#2}%
116     \def\f@series{#3}%
117     \def\f@shape{#4}%
118     \def\exfs@tempa{#5}}
119 \endgroup
120 \def\exfs@base@family{\expandafter\exfs@get@base\f@family\@nil}
121 \DeclareRobustCommand{\ornament}[1]{%
122   \expandafter\ifx\csname\exfs@base@family @orn\@roman#1\endcsname\relax
123     \PackageWarning{nfssect}{%
124       Ornament #1 undefined for font family '\exfs@base@family'\MessageBreak
125       Setting debug mark}%
126     \rule{1ex}{1ex}%
127   \else
128     \begingroup
129     \edef\exfs@tempb{\csname\exfs@base@family @orn\@roman#1\endcsname}%
130     \expandafter\expandafter\expandafter\exfs@split@orndef
131     \expandafter\string\exfs@tempb\@nil
132     \selectfont\char\exfs@tempa
133   \endgroup
134   \fi}

```

`\nfssectcfr@MT@Hook` BEGIN add microtype hooks Partly from microtype docs; partly from MinionPro package
`\Microtype@Hook`

```

135 \def\nfssectcfr@MT@Hook{%
136   \DeclareMicrotypeVariants*{2,2d,2dj,2j,dj,e,h,l}% is this necessary or
       would the previous line be enough?
137 }
138 \@ifpackageloaded{microtype}{%
139   \PackageWarning{nfssect-cfr}{%
140     You have loaded me (or a font support package which loads me)\MessageBreak
141     after loading microtype, but microtype should be loaded after\MessageBreak
142     all font defaults have been setup}%
143   \nfssectcfr@MT@Hook
144 }{%
145   \@ifundefined{Microtype@Hook}{%
146     \let\Microtype@Hook\nfssectcfr@MT@Hook}{% MinionPro has \global before
       this
147     \g@addto@macro\Microtype@Hook{\nfssectcfr@MT@Hook}}}

```

END

6.2 NNFSS

This code was written for the *current* (New) New Font Selection Scheme (2020–). It should not generally be loaded on older kernels.

`nfssect-cfr-nnfss` (*pkg.*)

```

148 \NeedsTeXFormat{LaTeX2e}
149 \RequirePackage{svn-prov}
150 \ProvidesPackageSVN[\filebase-nnfss.sty]{\Id: nfssect-cfr.dtx 10366 2024-09-18
       14:25:21Z cfrees $}[v1.0 \revinfo{} extended New New Font Selection Scheme
       (NNFSS) based on 2003/03/14 v1.2 Experimental NFSS Extensions]
151 \DefineFileInfoSVN

```

END added

`\exfs@tempa` Scratch variables.

```

\exfs@tempb
\exfs@tempf 152 \newcommand*{\exfs@tempa}{}
153 \newcommand*{\exfs@tempb}{}

```

:end-added BEGIN added (cfr): extra variable (`\exfs@tempf`)

```

154 \newcommand*{\exfs@tempf}{}

```

We want to track cases where missing fonts get defined into existence. To do this we define an additional macro each time `\wrong@fontshape` is called. This is based on two proposals by Max Chernoff, but the implementation is different.

```

155 \ExplSyntaxOn
156 \hook_gput_code:nnn { cmd/wrong@fontshape/before } { . }
157 {
158   \global\expandafter\expandafter\expandafter\let
159   \expandafter \csname exfs@fake@ \curr@fontshape\endcsname\relax
160 }
161 \ExplSyntaxOff

```


`\exfs@info` Custom logging

```

162 \newcommand \exfs@info[2][nfssect-cfr]{%
163   \ifexfs@debug
164     \PackageWarning{#1}{Info: #2}%
165   \else
166     \PackageInfo{#1}{#2}%
167   \fi
168 }
```

END added

`\exfs@normalise` BEGIN added (cfr): normalise

```

169 \newcommand* \exfs@normalise[1]{%
170   \ifcsname exfs@fake@\curr@fontshape\endcsname
171     \exfs@info{Current font '\curr@fontshape' is fake.\MessageBreak
172       Normalising}%
173   \expandafter\csname f@#1\endcsname{\csname #1default\endcsname}\selectfont
174   \ifcsname exfs@fake@\curr@fontshape\endcsname
175     \f@series{\seriesdefault}\f@shape{\shapedefault}\selectfont
176   \ifcsname exfs@fake@\curr@fontshape\endcsname
```

This might happen, I think, if we're in a swash family or specialist encoding where the default series and shape aren't available. All bets are off here so try to bale out as gracefully as we can.

```

177     \normalfont
178     \ifcsname exfs@fake@\curr@fontshape\endcsname
```

This definitely oughtn't to happen, though - things are really screwed up at this point - so error.

```

179     \PackageError{nfssect-cfr}{Default font appears to be fake!\MessageBreak
180       Switch \textbackslash normalfont yielded '\curr@fontshape'}
181     {This is highly unlikely, so the bug is probably in the phenomena\MessageBreak
182       rather than the noumena}%
183   \fi
184   \fi
185   \fi
186   \exfs@info{Normalised to '\curr@fontshape'}%
187 \else
188   \exfs@info{Current font '\curr@fontshape' appears real}%
189 \fi
190 }
```

END added

`\exfs@try@family` Modified from nfssect? Or modified from older nfssect-cfr?

```

191 \newcommand*{\exfs@try@family}[2][]{%
192   \let\exfs@tempa\relax
```

END added

trans: group is requisite here else L^AT_EX thinks the family real regardless

```
193 \begingroup % fel arall, bydd latex yn credu bod y family yn go iawn beth
    bynnag
```

(o leiaf bydd latex yn dweud felly)

```
194 \exfs@info{Trying Font family '\f@encoding/#2'}%
195 \fontfamily{#2}\try@load@fontshape
```

`\curr@fontshape` holds the target shape - not the current one - after an unsuccessful attempt to load **family** with `\try@load@fontshape`. This won't work for series/shape as `\curr@fontshape` holds the current one in that case

```
196 \expandafter\ifx\csname\curr@fontshape\endcsname\relax
197 \edef\exfs@tempa{#1}%
198 \ifx\exfs@tempa\@empty
199 \PackageWarning{nfssect}{%
200 Font family '\f@encoding/#2' not available\MessageBreak
201 Ignoring font switch}%
202 \else
203 \exfs@info[nfssect]{%
204 Font family '\f@encoding/#2' not available\MessageBreak
205 Font family '\f@encoding/#1' tried instead}%
206 \exfs@try@family{#1}%
207 \fi
208 \else
209 \exfs@info{Loading font family '\f@encoding/#2'}%
210 \gdef\exfs@tempa{\fontfamily{#2}\selectfont}%
211 \fi
212 \endgroup
213 \exfs@tempa}
```

`\exfs@try@series` BEGIN added (cfr)

```
214 \newcommand*{\exfs@try@series}[2][{}]{%
```

We don't hand instructions to the kernel unless we know they'll succeed b/c the results are too unpredictable under NNFSS.

Changing directly only produces usable results for series defined in the 'table' of font changes. But using higher level kernel interfaces for tests doesn't work because spurious fonts get defined, which only seem to exist. Theoretically, we might as well use the existing kernel's macros since we're already damned by reliance on internals anyway. But then everything needs disentangling. So it's easier to just adapt previous tests, even though it partially duplicates what the kernel does. (But it isn't the mess swash is ...).

```
215 \let\exfs@targetseries\relax
216 \edef\exfs@tempa{#2}%
217 \ifx\f@series\exfs@tempa\relax
218 \exfs@info{Current (\f@series) matches target (#2) series.\MessageBreak
219 Ignoring font switch}%
220 \else
221 \begingroup
222 \exfs@normalise{series}%
```

```

223 \edef\exfs@tempa{\f@encoding/\f@family/#2/\f@shape}%
224 \ifcsname \exfs@tempa\endcsname
225 \exfs@info{Switching series: \f@series\space -> #2}%
226 \gdef\exfs@targetseries{\fontseries{#2}\selectfont}%
227 \else
228 \edef\exfs@reserved{#1}%
229 \ifx\exfs@reserved\@empty
230 \PackageWarning{nfssect-cfr}{%
231 Font series '\f@encoding/\f@family/#2/\f@shape' not available\MessageBreak
232 Ignoring font change}%
233 \else
234 \PackageWarning{nfssect-cfr}{%
235 Font series '\f@encoding/\f@family/#2/\f@shape' not available\MessageBreak
236 Trying '\f@encoding/\f@family/#1/\f@shape'}%
237 \exfs@try@series{#1}%
238 \fi
239 \fi
240 \endgroup
241 \exfs@targetseries
242 \fi}

```

`\exfs@try@shapeshift` Attempt to leverage kernel's mechanism.

```

243 \def\exfs@try@shapeshift#1{%
244 \edef\exfs@targetshape{\csname #1default\endcsname}%
245 \ifx\f@shape\exfs@targetshape\relax
246 \exfs@info{Current (\f@shape) matches target (#1) shape.\MessageBreak
247 Ignoring font switch}%
248 \else
249 \not@math@alphabet\edef\exfs@targetshape\relax
250 \exfs@info{\f@shape\space -> \exfs@targetshape\MessageBreak
251 Trying \f@encoding/\f@family/\f@series/\exfs@targetshape}%

```

We *do* want the kernel's substitution mechanism here?

```

252 \fontshape{\exfs@targetshape}\selectfont
253 \fi}

```

`\exfs@swshape` Switching to swash is now far more complicated with (I presume) the attendant overhead, but the kernel's approach just won't work here⁹. This will become `\swshape` if `compat` isn't enabled.

```

254 \newcommand* \exfs@swshape{%
255 \let\exfs@targetsw\relax
256 \begingroup % angen neu beidio? angen - bendant!

```

Try kernel or configured default first so we get swash from current family etc. if available

⁹Implementing swash as a *shape* isn't workable for fonts I've packaged, so I've made no attempt to follow the kernel here as I do for small-caps italic etc. We're back to the single axis/multiple aspect problem which NFSS created by ignoring small-caps/italic and width/weight combinations. It may, in fact, be wrong-headed to follow the kernel *at all* here. Perhaps it would be better to just provide the original implementation and some compatibility option for people who also need swash shapes in the same document?

```

257 \edef\exfs@tempa{\f@encoding/\f@family/\f@series/\swdefault}%
258 \edef\exfs@tempa@fake{\exfs@fake@\exfs@tempa}%
259 \ifcsname \exfs@tempa\endcsname
260 \ifcsname exfs@fake@\exfs@tempa\endcsname
261 \exfs@swfamily
262 \else
263 \gdef\exfs@targetsw{\fontshape{\swdefault}\selectfont}% kernel
264 \fi
265 \else
266 \exfs@swfamily
267 \fi
268 \endgroup
269 \ifx\exfs@targetsw\relax
270 \PackageWarning{nfssect-cfr}{%
271 Cannot find any route to swash.\MessageBreak
272 Are you sure one is available?}%
273 \else
274 \exfs@targetsw
275 \exfs@info{Switch to swash resulted in '\curr@fontshape'}%
276 \fi
277 }

```

\exfs@swfamily This is the guts of \exfs@swshape.

```

278 \newcommand* \exfs@swfamily{%
279 \let\exfs@targetsw\relax
280 \begingroup

```

Try nfssect-cfr family switch & our default or configured

```

281 \let\exfs@tempa\f@family
282 \exfs@merge@families{w}%
283 \ifx\exfs@tempa\f@family % try merge with current shape

```

Try switching to upright etc. first

```

284 \fontshape{n}%
285 \exfs@merge@families{w}% up & merge
286 \if\exfs@tempa\f@family

```

Try switching to \swshapedefault first

```

287 \fontshape\swshapedefault
288 \exfs@merge@families{w}% up & nfssect-cfr default/configured
289 \if\exfs@tempa\f@family

```

Use nfssect family switch & default or configured

```

290 \exf@try@family{\expandafter\exfs@get@base\f@family\@nil w}% nfssect
switch
291 \if\exfs@tempa\f@family % nfssect switch
292 \fontshape\swshapedefault\exfs@try@family{%
293 \expandafter\exfs@get@base\f@family\@nil w}% nfssect switch &
shape
294 \if\exfs@tempa\f@family
295 \relax % rhodd y ffdl yn y to (give up)

```

```

296         \else
297             \gdef\exfs@targetsw{%
298                 \fontshape\swshapedefault\expandafter\fontfamily{%
299                     \exfs@get@base\f@family\@nil w}\selectfont
300             }% nfssect switch & shape
301         \fi % nfssect switch & shape
302     \else
303         \gdef\exfs@targetsw{\expandafter\fontfamily{%
304             \exfs@get@base\f@family\@nil w}\selectfont
305         }% nfssect switch
306     \fi % nfssect switch
307 \else
308     \gdef\exfs@targetsw{%
309         \fontshape\swshapedefault\exfs@merge@families{w}%
310     }% up & nfssect-cfr default/configured
311 \fi % up & nfssect-cfr default/configured
312 \else
313     \gdef\exfs@targetsw{%
314         \fontshape{n}\exfs@merge@families{w}%
315     }% up & merge
316 \fi % up & merge
317 \else
318     \gdef\exfs@targetsw{\exfs@merge@families{w}}% merge with current shape
319 \fi % merge with current shape
320 \endgroup
321 }

```

END added

```

\exfs@get@base Utilities
\exfs@get@variants
\exfs@next 322 \def\exfs@get@base#1#2#3#4\@nil{#1#2#3}
\exfs@shift BEGIN added (cfr): more \exfs@ commands (get@variants, next, shift, first,
\exfs@first part, second)
\exfs@part
\exfs@second 323 \def\exfs@get@variants#1#2#3#4\@nil{#4}
324 \def\exfs@next#1#2\@nil{#1}
325 \def\exfs@shift#1#2\@nil{#2}
326 \def\exfs@first#1#2\@nil{#1}
327 \def\exfs@part#1#2\@nil{#2}
328 \def\exfs@second#1#2#3\@nil{#2}

```

\exfs@series@splitter Common method for dealing with weight and width.

```

329 \def\exfs@series@splitter#1{%
330     \edef\exfs@weight{\expandafter\exfs@first#1\@nil}%
331     \edef\exfs@width{\expandafter\exfs@shift#1\@nil}%

```

Two char width only or two char weight

```

332     \if\exfs@weight u\exfs@check@cx{u}%
333     \else\if\exfs@weight e\exfs@check@cx{e}%
334     \else\if\exfs@weight s\exfs@check@cx{s}%
335     \else\if\exfs@weight d\exfs@check@cx{d}%

```

```
336         \else\ifx\exfs@width\@empty % m dealt with elsewhere
```

Single character width

```
337         \if\exfs@weight c\def\exfs@width{c}\let\exfs@weight\@empty
338         \else\if\exfs@weight x\def\exfs@width{x}\let\exfs@weight\@empty
339         \fi\fi
340     \fi\fi\fi\fi\fi
341     \exfs@info{#1 -> \exfs@weight:\exfs@width;}%
342 }
```

\exfs@check@cx Auxiliary for \exfs@check@cx

```
343 \def\exfs@check@cx#1{%
344     \edef\exfs@tempa{\expandafter\exfs@first\exfs@width\@nil}%
345     \if\exfs@tempa c\edef\exfs@width{#1c}\let\exfs@weight\@empty
346     \else\if\exfs@tempa x\edef\exfs@width{#1x}\let\exfs@weight\@empty
347     \else\edef\exfs@weighta{%
348         \exfs@weight\exfs@tempa
349     }\let\exfs@weight\exfs@weighta
350     \edef\exfs@widtha{%
351         \expandafter\exfs@shift\exfs@width\@nil
352     }\let\exfs@width\exfs@widtha
353     \fi\fi
354 }
```

END added

```
\lnstyle Unmodified from nfssect. Anniffiniedig -> undefined in the kernel.
\osstyle
\infstyle 355 \DeclareRobustCommand{\lnstyle}{%% anniffiniedig
\instyle 356 \not@math@alphabet\lnstyle\relax
\style 357 \exfs@try@family{\expandafter\exfs@get@base\family\@nil}%
\swstyle 358 {\expandafter\exfs@get@base\family\@nil x}}
359 \DeclareRobustCommand{\osstyle}{%% anniffiniedig
360 \not@math@alphabet\osstyle\relax
361 \exfs@try@family{\expandafter\exfs@get@base\family\@nil j}}
362 \DeclareRobustCommand{\infstyle}{%% anniffiniedig
363 \not@math@alphabet\infstyle\relax
364 \exfs@try@family{\expandafter\exfs@get@base\family\@nil 0}}
365 \let\instyle\infstyle
366 \DeclareRobustCommand{\sustyle}{%% anniffiniedig
367 \not@math@alphabet\sustyle\relax
368 \exfs@try@family{\expandafter\exfs@get@base\family\@nil 1}}
369 \DeclareRobustCommand{\swstyle}{%% anniffiniedig
370 \not@math@alphabet\swstyle\relax
371 \exfs@try@family{\expandafter\exfs@get@base\family\@nil w}}
```

BEGIN added (cfr) - merge families.

NNFSS (unsurprisingly) does nothing here, so this is unproblematic. The following depends ****absolutely**** on ****complete**** adherence to berry names.

\ifexfs@added \exfs@merge@families is used in the macros recommended for switching the style
\exfs@merge@families of figures, activating swash and other variants etc.

```

372 \newif\ifexfs@added
373 \newcommand*\exfs@merge@families[1]{%
374   \edef\exfs@vartomerge{#1}%
375   \edef\exfs@variants{\expandafter\exfs@get@variants\f@family\@nil}%
376   \exfs@info{Trying to merge variants #1 and \exfs@variants}%
377   \edef\tempo{2j}%
378   \let\exfs@tempq\@empty
379   \def\exfs@tempg{}%
380   \exfs@addedfalse

```

Check whether there are variants - if not just use the requested addition.

```

381   \ifx\exfs@variants\@empty
382     \edef\exfs@tempq{\exfs@vartomerge}%
383     \exfs@addedtrue
384   \else
385     \gdef\set{0,1,2,a,d,e,f,h,j,l,p,q,s,t,v,w}% these are the variants to
      consider - the order here and in the font name is crucial
386     \ifx\tempo\exfs@vartomerge
387       \@for \xx:=\set \do {%

```

Check whether there are variants left - if not set the ‘next variant’ to empty

```

388       \ifx\exfs@variants\@empty
389         \let\exfs@nextvariant\@empty
390       \else

```

O/w get the next variant

```

391         \edef\exfs@nextvariant{\expandafter\exfs@next\exfs@variants\@nil}%
392       \fi

```

If the next variant is 2 or j, ignore it

```

393       \if\exfs@nextvariant 2%
394         \edef\exfs@variants{\expandafter\exfs@shift\exfs@variants\@nil}%
395       \fi
396       \if\exfs@nextvariant j% if the next variant is j, ignore it
397         \edef\exfs@variants{\expandafter\exfs@shift\exfs@variants\@nil}%
398       \fi

```

See if the current value is either 2 or j and add it if so and if needed

```

399       \if\xx 2%
400         \edef\exfs@tempg{\exfs@tempg\xx}%
401       \else
402         \if\xx j% if the current value is j, we’re done
403           \edef\exfs@tempq{\exfs@tempg\xx\exfs@variants}%
404           \let\exfs@variants\@empty
405           \exfs@addedtrue
406         \else

```

o/w see if the current value matches the next variant

```

407           \ifx\xx\exfs@nextvariant
408             \edef\exfs@tempg{\exfs@tempg\xx}%
409             \edef\exfs@variants{\expandafter\exfs@shift\exfs@variants\@nil}%

```

```

410         \fi
411     \fi
412 \fi
413 }%
414 \else
415     \@for \xx:=\set \do {%

```

Check whether there are variants left and, if not, add the addition if needed

```

416         \ifx\exfs@variants\@empty
417             \ifexfs@added
418             \else
419                 \edef\exfs@tempq{\exfs@tempg\exfs@vartomerge}%
420                 \exfs@addedtrue
421             \fi
422         \else

```

o/w get the next variant

```

423             \edef\exfs@nextvariant{\expandafter\exfs@next\exfs@variants\@nil}%

```

If the new token equals the next variant, combine whatever is saved in \exfs@tempg with whatever remains in \exfs@variants

```

424             \ifx\exfs@nextvariant\exfs@vartomerge
425                 \edef\exfs@tempq{\exfs@tempg\exfs@variants}%
426                 \exfs@addedtrue
427                 \let\exfs@variants\@empty
428             \else

```

o/w, if the current value matches the requested addition, add it in

```

429             \ifx\exfs@vartomerge\xx
430                 \edef\exfs@tempq{\exfs@tempg\xx\exfs@variants}%
431                 \exfs@addedtrue
432                 \let\exfs@variants\@empty
433             \else

```

o/w, if the current value matches the next variant, shift

```

434             \ifx\exfs@nextvariant\xx
435                 \edef\exfs@tempg{\exfs@tempg\xx}%
436                 \edef\exfs@variants{\expandafter\exfs@shift\exfs@variants\@nil}%
437             \fi
438         \fi
439     \fi
440 \fi
441 }%
442 \fi
443 \fi
444 \ifx\exfs@tempq\@empty
445     \PackageError{nfssect-cfr}{Something is wrong here. Ignoring font switching
command.}{}%
446 \else
447     \exfs@try@family{\expandafter\exfs@get@base\fontfamily\@nil \exfs@tempq}%
448 \fi
449 }

```


`\pstyle` Commands for switching to proportional and/or oldstyle figures. Compare `\ostyle` with `nfssex`'s `\osstyle` above. These macros (and the block which follows) all `\postyle` require merging Berry names but not unmerging.

```

450 \DeclareRobustCommand{\pstyle}{%% anniffiniedig proportional figures
451   \not@math@alphabet\pstyle\relax
452   \exfs@merge@families{2}}
453 \DeclareRobustCommand{\ostyle}{%% anniffiniedig oldstyle figures (cf. original
    osstyle above)
454   \not@math@alphabet\ostyle\relax
455   \exfs@merge@families{j}}
```

Combined command for proportional oldstyle

```

456 \DeclareRobustCommand{\postyle}{%% anniffiniedig
457   \not@math@alphabet\postyle\relax
458   \exfs@merge@families{2j}}
```

`\tistyle` These macros again require merging, but not unmerging, names.

```

\ltstyle 459 \DeclareRobustCommand{\tistyle}{%% anniffiniedig titling/display
\ofstyle 460   \not@math@alphabet\tistyle\relax
\altstyle 461   \exfs@merge@families{d}}
\regstyle
```

`\embossstyle` Note that this command is for use when the light version is a separate family rather than a weight variant (e.g. when you've got light, light bold etc. as well as regular weights)

```

\shstyle 462 \DeclareRobustCommand{\ltstyle}{%% anniffiniedig
\qtstyle 463   \not@math@alphabet\ltstyle\relax
          464   \exfs@merge@families{l}}
```

Let's hope there aren't any fonts with a light family **and** an outline/openface/blank version

```

465 \DeclareRobustCommand{\ofstyle}{%% anniffiniedig
466   \not@math@alphabet\ofstyle\relax
467   \exfs@merge@families{l}}
468 \DeclareRobustCommand{\altstyle}{%% anniffiniedig alternative style
469   \not@math@alphabet\altstyle\relax
470   \exfs@merge@families{a}}
471 \DeclareRobustCommand{\regstyle}{%% anniffiniedig 'regular' style
472   \not@math@alphabet\regstyle\relax
473   \exfs@try@family{\expandafter\exfs@get@base\family\@nil}}
474 \DeclareRobustCommand{\embossstyle}{%% anniffiniedig
475   \not@math@alphabet\embossstyle\relax
476   \exfs@merge@families{e}}
477 \DeclareRobustCommand{\ornamentalstyle}{%% anniffiniedig intended primarily
    for decorative initial fonts etc.
478   \not@math@alphabet\ornamentalstyle\relax
479   \exfs@merge@families{p}}
480 \DeclareRobustCommand{\qtstyle}{%% anniffiniedig quotation style (assumes
    sans)
481   \not@math@alphabet\qtstyle\relax
482   \sffamily
483   \exfs@merge@families{q}}
```

```

484 \DeclareRobustCommand{\shstyle}{%% anniffiniedig
485   \not@math@alphabet\shstyle\relax
486   \exfs@merge@families{h}}
487 \DeclareRobustCommand{\swashstyle}{%% anniffiniedig    an attempt to improve
      on \swstyle
488   \not@math@alphabet\swashstyle\relax
489   \exfs@merge@families{w}}

```

`\tmstyle` Macros to switch between monowidth and variable typewriter. These need to
`\tvstyle` unmerge before merging. We need to unmerge sans as well as the other kind of
typewriter.

```

490 \DeclareRobustCommand{\tmstyle}{%% anniffiniedig      monowidth typewriter
491   \not@math@alphabet\tmstyle\relax
492   \exfs@unmerge@families{s}%
493   \exfs@unmerge@families{v}%
494   \exfs@merge@families{t}}
495 \DeclareRobustCommand{\tvstyle}{%% anniffiniedig      variable width typewriter
496   \not@math@alphabet\tvstyle\relax
497   \exfs@unmerge@families{s}%
498   \exfs@unmerge@families{t}%
499   \exfs@merge@families{v}}

```

BEGIN added (cfr) - unmerge families

`\ifexfs@take` Define the macro needed to do the unmerges.
`\exfs@unmergefamilies`

```

500 \newif\ifexfs@take
501 \newcommand*\exfs@unmerge@families[1]{%
502   \edef\exfs@tempf{#1}%
503   \edef\tempa{\expandafter\exfs@get@variants\f@family\@nil}%
504   \let\exfs@tempq\@empty
505   \edef\exfs@tempg{}%
506   \exfs@taketrue

```

Check whether there are variants - if not do nothing

```

507   \ifx\tempa\@empty
508     \edef\exfs@tempq{}%
509   \else

```

o/w go through the variants to find the one to delete

```

510     \loop

```

Get the next variant

```

511         \edef\exfs@tempn{\expandafter\exfs@next\tempa\@nil}%

```

See if the next variant is the thing we seek and, if so, eliminate it

```

512         \ifx\exfs@tempf\exfs@tempn
513           \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
514           \edef\exfs@tempq{\exfs@tempg\tempa}%
515           \exfs@takefalse

```

o/w save the next variant and move on if any variants remain

```

516     \else
517     \edef\exfs@tempg{\exfs@tempg\exfs@tempn}%
518     \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
519     \ifx\tempa\@empty% if there are no variants left, we're done
520     \edef\exfs@tempq{\exfs@tempg}%
521     \exfs@takefalse
522     \fi
523     \fi
524     \ifexfs@take % \fi yn \repeat
525     \repeat
526     \fi
527     \exfs@try@family{\expandafter\exfs@get@base\family\@nil \exfs@tempq}%
528 }

```

`\tstyle` These are simple unmerges, with no merging necessary.

```

\lstyle
529 \DeclareRobustCommand{\tstyle}{%% anniffiniedig tabular figures
530   \not@math@alphabet\tstyle\relax
531   \exfs@unmerge@families{2}}
532 \DeclareRobustCommand{\lstyle}{%% anniffiniedig lining figures (cf. command
    above)
533   \not@math@alphabet\lstyle\relax
534   \exfs@unmerge@families{j}}

```

`\tlstyle` Simple combinations for combined figure styles.

`\plstyle` Make a combined command for tabular lining

```

\tostyle
535 \DeclareRobustCommand{\tlstyle}{%% anniffiniedig
536   \lstyle\tstyle}

```

Proportional lining

```

537 \DeclareRobustCommand{\plstyle}{%% anniffiniedig
538   \lstyle\pstyle}

```

Tabular oldstyle ?!

```

539 \DeclareRobustCommand{\tostyle}{%% anniffiniedig
540   \ostyle\tstyle}

```

END added

`\sidefault` `si` is italic `sc`¹⁰. We use the original definition for the default and then set up rules `\sishape` for font shape changes which try `scit` and `scsl` before falling back to `si`.

```

541 \newcommand*{\sidefault}{si}% anniffiniedig

```

Well i brofi `si` & yna `scit` ond wn i ddim sut i wneud hon gyda'r stwff newydd I was going to deprecate the `si` macros, but the truth is they are much nicer than having to combine macros for those cases when you really do want precisely italic small-caps. They are also much more robust than somebody trying to force

¹⁰That is, italic small-caps *was* `si`. These days, things are more complicated

things with `\fontshape{<si>}\selectfont`, so, on reflection, it seems better to retain the interface, even if the implementation isn't as straightforward as I'd like.

```
542 \DeclareRobustCommand{\sishape}{%
543   \exfs@try@shapeshift{si}}
```

Kernel virtuals: `ulc` upper/lower case up upright. `nfssex` virtuals? or `nfssex-cfr` virtuals? or? It would be better to try `si`, `scit` and `scsl`, but that doesn't seem possible

```
544 \DeclareFontShapeChangeRule {n}{si}{scit}{scsl}% current; request (& trydedd);
    dewisiad cyntaf; ail ddewisiad
545 \DeclareFontShapeChangeRule {it}{si}{scit}{scsl}
546 \DeclareFontShapeChangeRule {sl}{si}{scit}{scsl}
547 \DeclareFontShapeChangeRule {sc}{si}{scit}{scsl}
```

Current shape is `si` => font support hasn't been updated

```
548 \DeclareFontShapeChangeRule {si}{sc} {si} {}
549 \DeclareFontShapeChangeRule {si}{it} {si} {}
550 \DeclareFontShapeChangeRule {si}{sl} {scsl} {si}
551 \DeclareFontShapeChangeRule {si}{scit} {scit} {si}% rhag ofn?!
552 \DeclareFontShapeChangeRule {si}{ulc} {it} {}
553 \DeclareFontShapeChangeRule {si}{up}{sc}{}

554 \ExplSyntaxOn
```

The need for overwriting can be avoided by changing the `.fd` files, but `compat` supports packages I don't know about¹¹ ...

```
555 \bool_if:NT \g__exfs_compat_bool
556 {
557   \DeclareFontShapeChangeRule {it}{sc}{si}{scsl}
558   \DeclareFontShapeChangeRule {sl}{sc}{scsl}{si}
```

Gofyn am italic etc. | Ask about italic etc. Sylwadau tebygol yma ... | Like comments here ...

```
559   \DeclareFontShapeChangeRule {sc}{it} {si} {scsl}
560   \DeclareFontShapeChangeRule {sc}{sl} {scsl} {si}
561   \DeclareFontShapeChangeRule {scsl}{it} {si} {scsl}
562 }
563 \ExplSyntaxOff
```

```
564 \DeclareFontShapeChangeRule {ui}{sc}{scit}{scsl}
565 \DeclareFontShapeChangeRule {ui}{scsl}{scsl}{scit}
566 \DeclareFontShapeChangeRule {ui}{it}{it}{ui}
567 \DeclareFontShapeChangeRule {ui}{ri}{ri}{ui}
568 \DeclareFontShapeChangeRule {ui}{up}{n}{}
569 \DeclareFontShapeChangeRule {ui}{ulc}{ui}{}

570 \DeclareFontShapeChangeRule {ri}{sc}{scit}{scsl}
571 \DeclareFontShapeChangeRule {ri}{scsl}{scsl}{scit}
572 \DeclareFontShapeChangeRule {ri}{it}{it}{ri}
573 \DeclareFontShapeChangeRule {ri}{ui}{ui}{ri}
```

¹¹Or haven't published? I haven't actually tried it with those.

Kernel virtuals: ulc upper/lower case up upright

```
574 \DeclareFontShapeChangeRule {ri}{up}{n}{}
575 \DeclareFontShapeChangeRule {ri}{ulc}{ri}{}

```

nfssexst virtuals? or nfssexst-cfr virtuals? or?

```
576 \DeclareFontShapeChangeRule {ol}{sc}{scol}{} % <- seiliedig ar nfssexst-cfr-nfss.sty
\scshape
577 \DeclareFontShapeChangeRule {ol}{ulc}{ol}{}
578 \DeclareFontShapeChangeRule {ol}{up}{ol}{}

579 \DeclareFontShapeChangeRule {scol}{sc}{scol}{}
580 \DeclareFontShapeChangeRule {scol}{ulc}{ol}{}
581 \DeclareFontShapeChangeRule {scol}{up}{scol}{}

582 \DeclareFontShapeChangeRule {u}{sc}{su}{} % <- seiliedig ar nfssexst-cfr-nfss.sty
\scshape
583 \DeclareFontShapeChangeRule {su}{ulc}{u}{}
584 \DeclareFontShapeChangeRule {sc}{u}{su}{}
585 \DeclareFontShapeChangeRule {su}{u}{su}{}
586 \DeclareFontShapeChangeRule {su}{sc}{su}{}

```

BEGIN added (cfr)

cfr: is this how outline shapes should be handled?

```
\oldefault Outline
\olshape
\scoldefault 587 \newcommand*{\oldefault}{ol}% anniffiniedig
\scshape 588 \DeclareRobustCommand{\olshape}{%% anniffiniedig
589 \exfs@try@shapeshift{ol}}
590 \newcommand*{\scoldefault}{scol}% anniffiniedig
591 \DeclareRobustCommand{\scolshape}{%% anniffiniedig
592 \exfs@try@shapeshift{scol}}

```

```
\udefault Underlined?? Fudge <- ??
\ushape
\scodefault 593 \newcommand*{\udefault}{u}% anniffiniedig
\scushape 594 \DeclareRobustCommand{\ushape}{%% anniffiniedig
595 \exfs@try@shapeshift{u}}
596 \newcommand*{\scodefault}{su}% anniffiniedig
597 \DeclareRobustCommand{\scushape}{%% anniffiniedig
598 \exfs@try@shapeshift{scu}}

```

```
\uidefault Upright and reverse italic
\uishape
\ridefault 599 \newcommand*{\uidefault}{ui}% anniffiniedig
\rishape 600 \DeclareRobustCommand{\uishape}{% anniffiniedig
601 \exfs@try@shapeshift{ui}}

```

Can i do this for reverse italic?

```
602 \newcommand*{\ridefault}{ri}% anniffiniedig
603 \DeclareRobustCommand{\rishape}{% anniffiniedig
604 \exfs@try@shapeshift{ri}}

```

END added BEGIN added (cfr) - merge width changes into series

`\exfs@merge@width` Previously dependent on incorrect series names.

```
605 \newcommand*{\exfs@merge@width}[1]{%
```

cfr-added

Dibynnodd y côd gwreiddiol ar `*mc*` etc. & r'odd hynny'n anghywir

Instead of merging or unmerging cyclically, which means keeping track of everything, we split the current series (which requires some juggling, but hopefully less) and use the results.

```
606 \exfs@series@splitter{\f@series}%
607 \edef\exfs@temph{#1}%
608 \if\exfs@temph m\ifx\exfs@weight\@empty\else\let\exfs@temph\@empty\fi\fi
609 \if\exfs@weight m\ifx\exfs@temph\@empty\else\let\exfs@weight\@empty\fi\fi
610 \edef\exfs@series{\exfs@weight\exfs@temph}%
```

end cfr-added

```
611 \exfs@info{Trying \exfs@series}%
612 \exfs@try@series{\exfs@series}}
```

`\regwidth` ‘Regular’ width requires conditionally adding ‘m’.

```
613 \DeclareRobustCommand{\regwidth}{%% anniffiniedig
614 \not@math@alphabet\regwidth\relax}
```

cfr altered

```
615 \exfs@merge@width{m}}
```

`\nwdefault` Condensed widths.

```
\nwwidth
\cddefault 616 \newcommand*{\nwdefault}{c}% anniffiniedig
\cdwidth 617 \DeclareRobustCommand{\nwwidth}{%% anniffiniedig ond rheolau
618 \not@math@alphabet\nwwidth\relax
\ecdefault 619 \exfs@merge@width{\nwdefault}}% neu \exfs@try@series ?
\ecwidth 620 \newcommand*{\cddefault}{c}% anniffiniedig
\ucdefault 621 \DeclareRobustCommand{\cdwidth}{%% anniffiniedig ond rheolau
622 \not@math@alphabet\cdwidth\relax
623 \exfs@merge@width{\cddefault}}% neu \exfs@try@series ?
624 \newcommand*{\ecdefault}{ec}% anniffiniedig
625 \DeclareRobustCommand{\ecwidth}{%% anniffiniedig ond rheolau
626 \not@math@alphabet\ecwidth\relax
627 \exfs@merge@width{\ecdefault}}% neu \exfs@try@series ?
628 \newcommand*{\ucdefault}{uc}% anniffiniedig
629 \DeclareRobustCommand{\ucwidth}{%% anniffiniedig
630 \not@math@alphabet\ucwidth\relax
631 \exfs@merge@width{\ucdefault}}
```

`\etdefault` Extended/expanded widths.

```
\etwidth
\epdefault 632 \newcommand*{\etdefault}{x}% anniffiniedig
\epwidth
\exdefault
\exwidth
\uxdefault
\uxwidth
```

```

633 \DeclareRobustCommand{\etwidth}{%% anniffiniedig
634 \not@math@alphabet\etwidth\relax
635 \exfs@merge@width{\etdefault}}% neu \exfs@try@series ?
636 \newcommand*{\epdefault}{x}% anniffiniedig
637 \DeclareRobustCommand{\epwidth}{%% anniffiniedig ond rheolau
638 \not@math@alphabet\epwidth\relax
639 \exfs@merge@width{\epdefault}}% neu \exfs@try@series ?
640 \newcommand*{\exdefault}{ex}% anniffiniedig
641 \DeclareRobustCommand{\exwidth}{%% anniffiniedig
642 \not@math@alphabet\exwidth\relax
643 \exfs@merge@width{\exdefault}}
644 \newcommand*{\uxdefault}{ux}% anniffiniedig
645 \DeclareRobustCommand{\uxwidth}{%% anniffiniedig
646 \not@math@alphabet\uxwidth\relax
647 \exfs@merge@width{\uxdefault}}

```

\mdwdefault Medium.

```

\mdwidth
648 \newcommand*\mdwdefault{m}
649 \DeclareRobustCommand{\mdwidth}{%% anniffiniedig
650 \not@math@alphabet\mdwidth\relax
651 \exfs@merge@width{\mdwdefault}}

```

Posibl ond bydda i'n colli achosion yn siwr. | Possible but I'd lose cases for sure.
 Hefyd hoffwn i ddim dyfalu pa rheolau y bydden nhw eu dewis. | Also I wouldn't
 like to guess which rules they'll choose.

BEGIN added (cfr) merge weight changes into series

\exfs@merge@weight The pay off for setting up series splitting is that we can reuse the method here
 and, as in the case of width, the definition is greatly simplified¹².

```

652 \newcommand*{\exfs@merge@weight}[1]{%
653 \exfs@series@splitter{\f@series}%

```

Save current series so we can test for change

```

654 \let\exfs@tempg\f@series
655 \def\exfs@temph{#1}%
656 \if\exfs@temph m\relax
657 \ifx\exfs@width\empty\relax
658 \else
659 \let\exfs@temph\empty
660 \fi
661 \fi
662 \edef\exfs@series{\exfs@temph\exfs@width}%
663 \ifx\exfs@temph\exfs@series
664 \exfs@info{Trying \exfs@series}%
665 \exfs@try@series{\exfs@series}%
666 \else
667 \exfs@info{Trying \exfs@series, favouring \exfs@weight}%
668 \exfs@try@series[\exfs@weight]{\exfs@series}% assume user wants to change
weight even if this changes back to the default width
669 \fi}

```

¹²At least if you don't look at the splitter code.

`\mbdefault` Ref.: sources2e.pdf and/or stripped code in base.

`\mbweight` The annotation ‘anniffiniedig’ indicates the macro is *not* defined by the kernel as of 2024. The addition ‘ond rheolau’ means there are nonetheless relevant rules.

`\bddefault` The comment ‘dim byd i’w gael eu wneud yma’ indicates that defining the default is now sufficient and no additional font switch or text command is required.

`\bfweight`

`\bdweight`

```

670 \newcommand*{\mbdefault}{sb}% dim byd i’w gael ei wneud yma
671 \DeclareRobustCommand{\mbweight}{%% anniffiniedig
672   \not@math@alphabet\mbweight\relax
673   \exfs@merge@weight{\mbdefault}}
674 \newcommand*{\bddefault}{b}% dim byd i’w gael ei wneud yma
675 \DeclareRobustCommand{\bfweight}{%% anniffiniedig
676   \not@math@alphabet\bfweight\relax
677   \exfs@merge@weight{\bddefault}}
678 \DeclareRobustCommand{\bdweight}{%% anniffiniedig
679   \not@math@alphabet\bdweight\relax
680   \exfs@merge@weight{\bddefault}}

```

`\mwdefault` Regular, medium, default are all irregular, exceptional, fraught.

`\mdweight`

```

681 \newcommand*\mwdefault{m}
682 \DeclareRobustCommand{\mdweight}{%% anniffiniedig
683   \not@math@alphabet\mdweight\relax
684   \exfs@merge@weight{\mwdefault}}

```

`\dbdefault` Heavy weights.

`\dbweight`

`\sbdefault`

`\sbweight`

`\ebdefault`

`\ebweight`

`\ubdefault`

`\ubweight`

```

685 \newcommand*{\dbdefault}{db}% anniffiniedig
686 \DeclareRobustCommand{\dbweight}{%% anniffiniedig
687   \not@math@alphabet\dbweight\relax
688   \exfs@merge@weight{\dbdefault}}
689 \newcommand*{\sbdefault}{sb}% anniffiniedig
690 \DeclareRobustCommand{\sbweight}{%% anniffiniedig ond rheolau
691   \not@math@alphabet\sbweight\relax
692   \exfs@merge@weight{\sbdefault}}% neu \exfs@try@series ?
693 \newcommand*{\ebdefault}{eb}% anniffiniedig
694 \DeclareRobustCommand{\ebweight}{%% anniffiniedig ond rheolau
695   \not@math@alphabet\ebweight\relax
696   \exfs@merge@weight{\ebdefault}}% neu \exfs@try@series ?
697 \newcommand*{\ubdefault}{ub}% anniffiniedig
698 \DeclareRobustCommand{\ubweight}{%% anniffiniedig ond rheolau
699   \not@math@alphabet\ubweight\relax
700   \exfs@merge@weight{\ubdefault}}% neu \exfs@try@series ?

```

`\lgdefault` Light weights.

`\lgweight`

`\eldefault`

`\elweight` Note - use this if light is a variant weight, rather than a separate family

`\uldefault`

`\ulweight`

```

701 \newcommand*{\lgdefault}{l}% anniffiniedig
702 \DeclareRobustCommand{\lgweight}{%% anniffiniedig ond rheolau
703   \not@math@alphabet\lgweight\relax
704   \exfs@merge@weight{\lgdefault}}% neu \exfs@try@series ?
705 \newcommand*{\eldefault}{el}% anniffiniedig

```



```

706 \DeclareRobustCommand{\elweight}{%% anniffiniedig ond rheolau
707 \not@math@alphabet\elweight\relax
708 \exfs@merge@weight{\eldefault}}% neu \exfs@try@series ?
709 \newcommand*{\uldefault}{ul}% anniffiniedig
710 \DeclareRobustCommand{\ulweight}{%% anniffiniedig ond rheolau
711 \not@math@alphabet\ulweight\relax
712 \exfs@merge@weight{\uldefault}}% neu \exfs@try@series ?

```

END added Original

`\dfshape` Something simpler.

```
713 \let\dfshape\normalshape
```

`\swshapedefault` cfr: be' i wneud am hwn?

```
714 \newcommand*{\swshapedefault}{\itdefault}
```

L^AT_EX ddim yn cynnwys `\swstyle` felly ...? | L^AT_EX doesn't include `\swstyle` so ...?

```

715 \ExplSyntaxOn
716 \hook_gput_code:nnn {begindocument}{.}
717 {% compatibility with original nfss \swshape

```

Note this doesn't affect `\swashstyle` or `\textswash`

`\swshape` Conditional definition. We overwrite the kernel's definition either way. The `compat` option determines only with what we overwrite it.

```

718 \bool_if:NTF \g__exfs_compat_bool
719 {
720   \DeclareRobustCommand{\swshape}
721   {
722     \not@math@alphabet\swshape\relax
723     \swstyle\fontshape\swshapedefault\selectfont
724   }
725   \PackageWarning{nfssext-cfr}{
726     Overwriting ~ kernel ~ definition ~ of ~ \swshape \space (compat)
727   }
728 }{
729   \DeclareRobustCommand \swshape
730   {
731     \not@math@alphabet\swshape\relax
732     \exfs@swshape
733   }
734   \PackageWarning{nfssext-cfr}{
735     Overwriting ~ kernel ~ definition ~ of ~ \swshape \space (new)
736   }
737 }

```

`\textin` Conditional definition.

```

738 \@ifpackageloaded{hyperref}{
739   \hook_gput_code:nnn {cmd/textin/before } { . }

```

```

740 {
741   \exfs@info{
742     Note ~ that ~ '\protect\textin' ~ is ~ defined ~ by ~ hyperref.\MessageBreak
743     Use ~ for ~ inferior ~ digits ~ will ~ yield ~ an\MessageBreak
744     undefined ~ command ~ error ~ in ~ document ~ font ~ encodings.\MessageBreak
745     Use ~ '\protect\textinf' ~ to ~ access ~ inferior ~ digits
746   }
747 }
748 }{
749   \DeclareTextFontCommand{\textin}{\infstyle}
750 }

751 }
752 \ExplSyntaxOff

```

`\textln` The annotation ‘anniffiniedig’ indicates the macro is *not* defined by the kernel as of 2024.

```

\textinf 753 \DeclareTextFontCommand{\textln}{\lnstyle}% anniffiniedig
\textsu 754 \DeclareTextFontCommand{\textos}{\osstyle}% anniffiniedig
\textsi 755 \DeclareTextFontCommand{\textinf}{\infstyle}% anniffiniedig
\textdf 756 \DeclareTextFontCommand{\textsu}{\sustyle}% anniffiniedig
757 \DeclareTextFontCommand{\textsi}{\sishape}% anniffiniedig
758 \DeclareTextFontCommand{\textdf}{\dfshape}% anniffiniedig

```

`\textsw` is already defined on newer kernels with essentially the same meaning as `nfssect` originally gave it, so we remove the definition here. However, the redefinition of `\swshape` means `\textsw` is effectively redefined, so the kernel definition is only technically retained.

BEGIN added (cfr)

```

\textti Families
\textlt
\textof 759 \DeclareTextFontCommand{\textti}{\tistyle}% anniffiniedig
\textalt 760 \DeclareTextFontCommand{\textlt}{\ltstyle}% anniffiniedig
761 \DeclareTextFontCommand{\textof}{\ofstyle}% anniffiniedig % open-face
\textreg (or outline or blank) style
\emboss 762 \DeclareTextFontCommand{\textalt}{\altstyle}% anniffiniedig % alternative
\textorn style
\textqt 763 \DeclareTextFontCommand{\textreg}{\regstyle}% anniffiniedig % ‘regular’
\textsh style
\texttm 764 \DeclareTextFontCommand{\emboss}{\embossstyle}% anniffiniedig
\texttv 765 \DeclareTextFontCommand{\textorn}{\ornamentalstyle}% anniffiniedig % intended
primarily for decorative initials etc.
766 \DeclareTextFontCommand{\textqt}{\qtstyle}% anniffiniedig
767 \DeclareTextFontCommand{\textsh}{\shstyle}% anniffiniedig % shadowed
style
768 \DeclareTextFontCommand{\texttm}{\tmstyle}% anniffiniedig
769 \DeclareTextFontCommand{\texttv}{\tvstyle}% anniffiniedig

```

`\textl` Families - figures

```

\texto 770 \DeclareTextFontCommand{\textl}{\lstyle}% anniffiniedig
\textp 771 \DeclareTextFontCommand{\texto}{\ostyle}% anniffiniedig
\textt

```

```

\textpl
\textpo
\texttl
\textto

```

```

772 \DeclareTextFontCommand{\textp}{\pstyle}% anniffiniedig
773 \DeclareTextFontCommand{\textt}{\tstyle}% anniffiniedig
774 \DeclareTextFontCommand{\textpl}{\plstyle}% anniffiniedig
775 \DeclareTextFontCommand{\textpo}{\postyle}% anniffiniedig
776 \DeclareTextFontCommand{\texttl}{\tlstyle}% anniffiniedig
777 \DeclareTextFontCommand{\textto}{\tostyle}% anniffiniedig

\textol Shapes
\textswash
\textu 778 \DeclareTextFontCommand{\textol}{\olshape}% anniffiniedig % outline
\textscu 779 \DeclareTextFontCommand{\textswash}{\swashstyle}% anniffiniedig % an attempt
to improve on \textsw
\textui 780 \DeclareTextFontCommand{\textu}{\ushape}% anniffiniedig % be' yw hwn?!
\textri <_ underlined?
781 \DeclareTextFontCommand{\textscu}{\scushape}% anniffiniedig
782 \DeclareTextFontCommand{\textui}{\uishape}% anniffiniedig % upright italic
783 \DeclareTextFontCommand{\textri}{\rishape}% anniffiniedig % reverse italic

\textnw Widths
\textcd
\textec 784 \DeclareTextFontCommand{\textnw}{\nwwidth}% anniffiniedig
\textuc 785 \DeclareTextFontCommand{\textcd}{\cdwidth}% anniffiniedig
\textet 786 \DeclareTextFontCommand{\textec}{\ecwidth}% anniffiniedig
\textep 787 \DeclareTextFontCommand{\textuc}{\ucwidth}% anniffiniedig
\textex 788 \DeclareTextFontCommand{\textet}{\etwidth}% anniffiniedig
\textux 789 \DeclareTextFontCommand{\textep}{\epwidth}% anniffiniedig
\textrw 790 \DeclareTextFontCommand{\textex}{\exwidth}% anniffiniedig
791 \DeclareTextFontCommand{\textux}{\uxwidth}% anniffiniedig
792 \DeclareTextFontCommand{\textrw}{\regwidth}% anniffiniedig

\textmb Weights
\textdb
\textbd 793 \DeclareTextFontCommand{\textmb}{\mbweight}% anniffiniedig
\textsb 794 \DeclareTextFontCommand{\textdb}{\dbweight}% anniffiniedig
\texteb 795 \DeclareTextFontCommand{\textbd}{\bdweight}% new?
\textub 796 \DeclareTextFontCommand{\textsb}{\sbweight}% anniffiniedig
\textlg 797 \DeclareTextFontCommand{\texteb}{\ebweight}% anniffiniedig
\textel 798 \DeclareTextFontCommand{\textub}{\ubweight}% anniffiniedig
\textul 799 \DeclareTextFontCommand{\textlg}{\lgweight}% anniffiniedig
800 \DeclareTextFontCommand{\textel}{\elweight}% anniffiniedig
801 \DeclareTextFontCommand{\textul}{\ulweight}% anniffiniedig

```

END added

BEGIN patch font initialisation for Latin Modern

Stop redefinition of bold if using Latin Modern as c_lm. Kernel default only blocks redefinition for l_m. Don't rely on c_{fr}-l_m internal macros as they may change without notice We don't need Dunhill, though, because it doesn't have bold of any kind. c_{fr}-l_m doesn't support using Quotation Sans as default or using e.g. serif as default sans, but there's nothing to stop somebody doing that so follow the kernel here even though it makes for a massive list¹³.

¹³I know this will go off the page when typeset, but I have no idea whether I can safely insert line breaks into the patch and I shall scream if I break this again. (Pun fully intended.)

```

802 \patchcmd{\init@series@setup}{\cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}{\cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}{}
803 \PackageWarning{nfssex-cfr}{%
804   Patching font initialisation macro for serif.%
805 }%
806 }%
807 \PackageWarning{nfssex-cfr}{%
808   Failed to patch font initialisation macro for serif.%
809 }%
810 }
811 \patchcmd{\init@series@setup}{\cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}{\cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}{}
812 \PackageWarning{nfssex-cfr}{%
813   Patching font initialisation macro for sans.%
814 }%
815 }%
816 \PackageWarning{nfssex-cfr}{%
817   Failed to patch font initialisation macro for sans.%
818 }%
819 }
820 \patchcmd{\init@series@setup}{\cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}{\cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}{}
821 \PackageWarning{nfssex-cfr}{%
822   Patching font initialisation macro for typewriter.%
823 }%
824 }%
825 \PackageWarning{nfssex-cfr}{%
826   Failed to patch font initialisation macro for typewriter.%
827 }%
828 }

```

END

6.3 NFSS

This code was written for the *old* New Font Selection Scheme (NFSS). It should not generally be loaded on current or recent kernels.

`nfssex-cfr-nfss` (*pkg.*)

```

829 \NeedsTeXFormat{LaTeX2e}
830 \RequirePackage{svn-prov}
831 \ProvidesPackageSVN[\filebase-nfss.sty]{$Id: nfssex-cfr.dtx 10366 2024-09-18
    14:25:21Z cfrees $}[v1.0 \revinfo{}] specially mangled by cfr; based on 2003/03/14
    v1.2 Experimental NFSS Extensions; for old NFSS]
832 \DefineFileInfoSVN

:cfr-added: use ifthen

833 \RequirePackage{ifthen}

:end-added

\exfs@tempa
\exfs@tempb
\exfs@tempf
834 \newcommand*\exfs@tempa{}
835 \newcommand*\exfs@tempb{}

```

:cfr-added: extra variable (\exfs@tempf)

```
836 \newcommand*{\exfs@tempf}{}
```

:end-added

\exfs@try@family

```
837 \newcommand*{\exfs@try@family}[2][{}]{%
838   \let\exfs@tempa\relax
839   \begingroup
840     \fontfamily{#2}\try@load@fontshape
841     \expandafter\ifx\csname\curr@fontshape\endcsname\relax
842     \edef\exfs@tempa{#1}%
843     \ifx\exfs@tempa\@empty
844       \PackageWarning{nfssexst}{%
845         Font family '\f@encoding/#2' not available\MessageBreak
846         Ignoring font switch}%
847     \else
848       \PackageInfo{nfssexst}{%
849         Font family '\f@encoding/#2' not available\MessageBreak
850         Font family '\f@encoding/#1' tried instead}%
851       \exfs@try@family{#1}%
852     \fi
853   \else
854     \gdef\exfs@tempa{\fontfamily{#2}\selectfont}%
855   \fi
856 \endgroup
857 \exfs@tempa}
```

\exfs@try@series :cfr-added \exfs@try@series

```
858 \newcommand*{\exfs@try@series}[2][{}]{%
859   \let\exfs@tempa\relax
860   \begingroup
861     \fontseries{#2}\try@load@fontshape
862     \expandafter\ifx\csname\curr@fontshape\endcsname\relax
863     \edef\exfs@tempa{#1}%
864     \ifx\exfs@tempa\@empty
865       \PackageWarning{nfssexst-cfr}{%
866         Font series '\f@encoding/\f@family/#2' not available\MessageBreak
867         Ignoring font switch}%
868     \else
869       \PackageInfo{nfssexst-cfr}{%
870         Font family '\f@encoding/\f@family/#2' not available\MessageBreak
871         Font family '\f@encoding/\f@family/#1' tried instead}%
872       \exfs@try@series{#1}%
873     \fi
874   \else
875     \gdef\exfs@tempa{\fontseries{#2}\selectfont}%
876   \fi
877 \endgroup
878 \exfs@tempa
879 }
```

:end-added

```

\exfs@get@base
\exfs@get@variants
\exfs@next
\exfs@shift
\exfs@first
\exfs@part
\exfs@second

880 \def\exfs@get@base#1#2#3#4\@nil{#1#2#3}
881 \def\exfs@get@variants#1#2#3#4\@nil{#4}
882 \def\exfs@next#1#2\@nil{#1}
883 \def\exfs@shift#1#2\@nil{#2}
884 \def\exfs@first#1#2\@nil{#1}
885 \def\exfs@part#1#2\@nil{#2}
886 \def\exfs@second#1#2#3\@nil{#2}

:cf-added:more \exfs@ commands (get@variants, next, shift, first, part, second)

:end-added

\lnstyle
\osstyle
\infstyle
\instyle
\sustyle
\swstyle

887 \DeclareRobustCommand{\lnstyle}{%
888   \not@math@alphabet\lnstyle\relax
889   \exfs@try@family[\expandafter\exfs@get@base\f@family\@nil]%
890   {\expandafter\exfs@get@base\f@family\@nil x}%
891 }
892 \DeclareRobustCommand{\osstyle}{%
893   \not@math@alphabet\osstyle\relax
894   \exfs@try@family{\expandafter\exfs@get@base\f@family\@nil j}%
895 \DeclareRobustCommand{\instyle}{%
896   \not@math@alphabet\instyle\relax
897   \exfs@try@family{\expandafter\exfs@get@base\f@family\@nil 0}%
898 \DeclareRobustCommand{\sustyle}{%
899   \not@math@alphabet\sustyle\relax
900   \exfs@try@family{\expandafter\exfs@get@base\f@family\@nil 1}%
901 \DeclareRobustCommand{\swstyle}{%
902   \not@math@alphabet\swstyle\relax
903   \exfs@try@family{\expandafter\exfs@get@base\f@family\@nil w}%

\exfs@merge@families :cf-added - merge families

904 \newcommand*\exfs@merge@families[1]{%
905   \edef\exfs@tempf{#1}%
906   \edef\tempa{\expandafter\exfs@get@variants\f@family\@nil}%
907   \edef\tempo{2j}%
908   \let\exfs@tempq\@empty
909   \def\exfs@tempg{}%
910   \newif\ifadded
911   \addedfalse

check whether there are variants - if not just use the requested addition

912 \ifx\tempa\@empty
913   \edef\exfs@tempq{\exfs@tempf}%
914   \addedtrue
915 \else
916   \gdef\set{0,1,2,a,d,e,f,h,j,l,p,q,s,t,v,w}% these are the variants to
consider - the order here and in the font name is crucial
917   \ifx\tempo\exfs@tempf
918     \@for \xx:=\set \do {%

```

check whether there are variants left - if not set the ‘next variant’ to empty

```

919      \ifx\tempa\@empty
920      \let\exfs@tempn\@empty
921      \else

```

o/w get the next variant

```

922      \edef\exfs@tempn{\expandafter\exfs@next\tempa\@nil}%
923      \fi
924      \edef\tempt{2}%
925      \edef\tempj{j}%

```

if the next variant is 2 or j, ignore it

```

926      \ifx\exfs@tempn\tempt
927      \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
928      \fi
929      \ifx\exfs@tempn\tempj % if the next variant is j, ignore it
930      \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
931      \fi

```

see if the current value is either 2 or j and add it if so and if needed

```

932      \ifx\tempt\xx
933      \edef\exfs@tempg{\exfs@tempg\xx}%
934      \else
935      \ifx\tempj\xx % if the current value is j, we’re done
936      \edef\exfs@tempq{\exfs@tempg\xx\tempa}%
937      \let\tempa\@empty
938      \addedtrue
939      \else

```

o/w see if the current value matches the next variant

```

940      \ifx\xx\exfs@tempn
941      \edef\exfs@tempg{\exfs@tempg\xx}%
942      \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
943      \fi
944      \fi
945      \fi
946      }%
947      \else
948      \@for \xx:=\set \do {%

```

check whether there are variants left and, if not, add the addition if needed

```

949      \ifx\tempa\@empty
950      \ifadded
951      \else
952      \edef\exfs@tempq{\exfs@tempg\exfs@tempf}%
953      \addedtrue
954      \fi
955      \else

```

o/w get the next variant

```

956      \edef\exfs@tempn{\expandafter\exfs@next\tempa\@nil}%

```

if the new token equals the next variant, combine whatever is saved in `\exfs@tempg` with whatever remains in `\tempa`

```

957          \ifx\exfs@tempn\exfs@tempf
958          \edef\exfs@tempq{\exfs@tempg\tempa}%
959          \addedtrue
960          \let\tempa\@empty
961          \else

```

o/w, if the current value matches the requested addition, add it in

```

962          \ifx\exfs@tempf\xx
963          \edef\exfs@tempq{\exfs@tempg\xx\tempa}%
964          \addedtrue
965          \let\tempa\@empty
966          \else

```

o/w, if the current value matches the next variant, shift

```

967          \ifx\exfs@tempn\xx
968          \edef\exfs@tempg{\exfs@tempg\xx}%
969          \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
970          \fi
971          \fi
972          \fi
973          \fi
974      }%
975  \fi
976  \fi
977  \ifx\exfs@tempq\@empty
978    \PackageError{nfssex-cfr}{Something is wrong here. Ignoring font switching
    command.}{}%
979  \else
980    \exfs@try@family{\expandafter\exfs@get@base\fontfamily\@nil \exfs@tempq}%
981    \fi
982 }

```

```

\pstyle
\ostyle
\postyle
\tistyle
983 \DeclareRobustCommand{\pstyle}{% proportional figures
984   \not@math@alphabet\pstyle\relax
985   \exfs@merge@families{2}}
986 \DeclareRobustCommand{\tistyle}{% titling/display
987   \not@math@alphabet\tistyle\relax
988   \exfs@merge@families{d}}
989 \DeclareRobustCommand{\ostyle}{% oldstyle figures (cf. original osstyle
    above)
990   \not@math@alphabet\ostyle\relax
991   \exfs@merge@families{j}}

```

combined command for proportional oldstyle

```

992 \DeclareRobustCommand{\postyle}{%
993   \not@math@alphabet\postyle\relax
994   \exfs@merge@families{2j}}

```



```

\ltstyle note that this command is for use when the light version is a separate family rather
\ofstyle than a weight variant (e.g. when you've got light, light bold etc. as well as regular
\altstyle weights)
\regstyle
\embossstyle 995 \DeclareRobustCommand{\ltstyle}{%
\ornamentalstyle 996 \not@math@alphabet\ltstyle\relax
\swashstyle 997 \exfs@merge@families{l}}
\shstyle let's hope there aren't any fonts with a light family *and* an outline/openface/blank
\qtstyle version

998 \DeclareRobustCommand{\ofstyle}{%
999 \not@math@alphabet\ofstyle\relax
1000 \exfs@merge@families{l}}
1001 \DeclareRobustCommand{\altstyle}{% alternative style
1002 \not@math@alphabet\altstyle\relax
1003 \exfs@merge@families{a}}
1004 \DeclareRobustCommand{\regstyle}{% 'regular' style
1005 \not@math@alphabet\regstyle\relax
1006 \exfs@try@family{\expandafter\exfs@get@base\family\@nil}}
1007 \DeclareRobustCommand{\embossstyle}{%
1008 \not@math@alphabet\embossstyle\relax
1009 \exfs@merge@families{e}}
1010 \DeclareRobustCommand{\ornamentalstyle}{% intended primarily for decorative
initial fonts etc.
1011 \not@math@alphabet\ornamentalstyle\relax
1012 \exfs@merge@families{p}}
1013 \DeclareRobustCommand{\qtstyle}{% quotation style (assumes sans)
1014 \not@math@alphabet\qtstyle\relax
1015 \sffamily
1016 \exfs@merge@families{q}}
1017 \DeclareRobustCommand{\shstyle}{%
1018 \not@math@alphabet\shstyle\relax
1019 \exfs@merge@families{h}}
1020 \DeclareRobustCommand{\swashstyle}{% an attempt to improve on \swstyle
1021 \not@math@alphabet\swashstyle\relax
1022 \exfs@merge@families{w}}

\tmstyle Macros to switch between monowidth and variable typewriter. These need to
\tvstyle unmerge before merging. We need to unmerge sans as well as the other kind of
typewriter.

1023 \DeclareRobustCommand{\tmstyle}{% monowidth typewriter
1024 \not@math@alphabet\tmstyle\relax
1025 \exfs@unmerge@families{s}%
1026 \exfs@unmerge@families{v}%
1027 \exfs@merge@families{t}}
1028 \DeclareRobustCommand{\tvstyle}{% variable width typewriter
1029 \not@math@alphabet\tvstyle\relax
1030 \exfs@unmerge@families{s}%
1031 \exfs@unmerge@families{t}%
1032 \exfs@merge@families{v}}

```

`\exfs@unmerge@families` :cfr-added - unmerge families

```

1033 \newcounter{taken}%
1034 \newcommand*\exfs@unmerge@families[1]{%
1035   \edef\exfs@tempf{#1}%
1036   \edef\tempa{\expandafter\exfs@get@variants\f@family\@nil}%
1037   \let\exfs@tempq\@empty
1038   \edef\exfs@tempg{}%
1039   \setcounter{taken}{0}%

check whether there are variants - if not do nothing

1040   \ifx\tempa\@empty
1041     \edef\exfs@tempq{}%
1042   \else

o/w go through the variants to find the one to delete

1043     \whiledo{\value{taken}<1}{%

get the next variant

1044       \edef\exfs@tempn{\expandafter\exfs@next\tempa\@nil}%

see if the next variant is the thing we seek and, if so, eliminate it

1045       \ifx\exfs@tempf\exfs@tempn
1046         \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
1047         \edef\exfs@tempq{\exfs@tempg\tempa}%
1048         \stepcounter{taken}%

o/w save the next variant and move on if any variants remain

1049       \else
1050         \edef\exfs@tempg{\exfs@tempg\exfs@tempn}%
1051         \edef\tempa{\expandafter\exfs@shift\tempa\@nil}%
1052         \ifx\tempa\@empty% if there are no variants left, we're done
1053           \edef\exfs@tempq{\exfs@tempg}%
1054           \stepcounter{taken}%
1055         \fi
1056       \fi
1057     }%
1058   \fi
1059   \exfs@try@family{\expandafter\exfs@get@base\f@family\@nil \exfs@tempq}%
1060 }

\tstyle
\lstyle
1061 \DeclareRobustCommand{\tstyle}{% tabular figures
1062   \not@math@alphabet\tstyle\relax
1063   \exfs@unmerge@families{2}}
1064 \DeclareRobustCommand{\lstyle}{% lining figures (cf. command above)
1065   \not@math@alphabet\lstyle\relax
1066   \exfs@unmerge@families{j}}

\tlstyle make a combined command for tabular lining
\plstyle
\tostyle
1067 \DeclareRobustCommand{\tlstyle}{%
1068   \lstyle\tstyle}

```

```

proportional lining

1069 \DeclareRobustCommand{\plstyle}{%
1070   \lstyle\pstyle}

tabular oldstyle ?!

1071 \DeclareRobustCommand{\tostyle}{%
1072   \ostyle\tstyle}

\sidefault :end-added si is italic sc
  \sishape
1073 \newcommand*{\sidefault}{si}
1074 \DeclareRobustCommand{\sishape}{%
1075   \not@math@alphabet\sishape\relax
1076   \fontshape\sidefault\selectfont}

\oldefault :cfr-added - is this how outline shapes should be handled?
  \olshape
1077 \newcommand*{\oldefault}{ol}
\scoldefault 1078 \DeclareRobustCommand{\olshape}{%
  \scolshape 1079   \not@math@alphabet\olshape\relax
1080   \fontshape\oldefault\selectfont}
1081 \newcommand*{\scoldefault}{scol}
1082 \DeclareRobustCommand{\scolshape}{%
1083   \not@math@alphabet\scolshape\relax
1084   \fontshape\scoldefault\selectfont}

\udefault :fudge
  \ushape
1085 \newcommand*{\udefault}{u}
\scudefault 1086 \DeclareRobustCommand{\ushape}{%
  \scushape 1087   \not@math@alphabet\ushape\relax
1088   \fontshape\udefault\selectfont}
1089 \newcommand*{\scudefault}{su}
1090 \DeclareRobustCommand{\scushape}{%
1091   \not@math@alphabet\scushape\relax
1092   \fontshape\scudefault\selectfont}

\uidefault :upright italic
  \uishape
1093 \newcommand*{\uidefault}{ui}
\ridefault 1094 \DeclareRobustCommand{\uishape}{%
  \rishape 1095   \not@math@alphabet\uishape\relax
1096   \fontshape\uidefault\selectfont}

:can i do this for reverse italic?

1097 \newcommand*{\ridefault}{ri}
1098 \DeclareRobustCommand{\rishape}{%
1099   \not@math@alphabet\rishape\relax
1100   \fontshape\ridefault\selectfont}

:end-added

```

`\exfs@merge@shape`

```

1101 \newcommand*{\exfs@merge@shape}[3]{%
1102   \edef\exfs@tempa{#1}%
1103   \edef\exfs@tempb{#2}%
1104   \ifx\f@shape\exfs@tempb
1105     \expandafter\ifx\csname\f@encoding/\f@family/\f@series/#3\endcsname\relax
1106     \else
1107       \edef\exfs@tempa{#3}%
1108     \fi
1109   \fi
1110   \fontshape{\exfs@tempa}\selectfont}

```

`\exfs@font@width` :cfr-added - merge width changes into series

```

1111 \newcommand*{\exfs@font@width}{%
1112   \edef\exfs@tempf{\expandafter\exfs@first\f@series\@nil }%
1113   \edef\exfs@temppart{\expandafter\exfs@part\f@series\@nil }%
1114   \ifx\exfs@temppart\@empty
1115     \def\exfs@width{}
1116   \else
1117     \edef\exfs@temps{\expandafter\exfs@second\f@series\@nil }%
1118     \ifx\exfs@temps{b}
1119       \edef\exfs@width{\expandafter\exfs@part\exfs@temps\@nil }%
1120     \else
1121       \ifx\exfs@temps{l}
1122         \edef\exfs@width{\expandafter\exfs@part\exfs@temps\@nil }%
1123       \else
1124         \edef\exfs@width{\exfs@temppart}%
1125       \fi
1126     \fi
1127   \fi
1128   \exfs@width
1129 }

```

`\exfs@merge@width`

```

1130 \newcommand*{\exfs@merge@width}[1]{%
1131   \edef\exfs@temph{#1}%
1132   \edef\exfs@tempf{\expandafter\exfs@first\f@series\@nil }%
1133   \edef\exfs@temppart{\expandafter\exfs@part\f@series\@nil }%
1134   \def\tempb{b}%
1135   \def\templ{l}%
1136   \ifx\exfs@temppart\@empty
1137     \def\exfs@series{\expandafter\exfs@tempf\exfs@temph}%
1138   \else
1139     \edef\exfs@temps{\expandafter\exfs@second\f@series\@nil }%
1140     \ifx\exfs@temps\tempb
1141       \def\exfs@series{\expandafter\exfs@tempf\exfs@temps\exfs@temph}%
1142     \else
1143       \ifx\exfs@temps\templ
1144         \def\exfs@series{\expandafter\exfs@tempf\exfs@temps\exfs@temph}%
1145       \else
1146         \def\exfs@series{\expandafter\exfs@tempf\exfs@temph}%
1147       \fi

```

```

1148     \fi
1149   \fi
1150   \exfs@try@series{\exfs@series}%
1151 }
1152 %^A \fontseries\exfs@series\selectfont}

```

\exfs@unmerge@width

```

1153 \newcommand*{\exfs@unmerge@width}{%
1154   \edef\exfs@tempf{\expandafter\exfs@first\f@series\@nil }%
1155   \edef\exfs@temppart{\expandafter\exfs@part\f@series\@nil }%
1156   \def\tempb{b}%
1157   \def\templ{l}%
1158   \ifx\exfs@temppart\@empty
1159     \def\exfs@series{\expandafter\exfs@tempf}%
1160   \else
1161     \edef\exfs@temps{\expandafter\exfs@second\f@series\@nil }%
1162     \ifx\exfs@temps\tempb
1163       \def\exfs@series{\expandafter\exfs@tempf\exfs@temps}%
1164     \else
1165       \ifx\exfs@temps\templ
1166         \def\exfs@series{\expandafter\exfs@tempf\exfs@temps}%
1167       \else
1168         \def\exfs@series{\expandafter\exfs@tempf}%
1169       \fi
1170     \fi
1171   \fi
1172   \exfs@try@series{\exfs@series}%
1173 }

```

\regwidth

```

1174 \DeclareRobustCommand{\regwidth}{%
1175   \not@math@alphabet\regwidth\relax
1176   \exfs@unmerge@width}

```

\nwdefault

\nwwidth

```

1177 \newcommand*{\nwdefault}{c}
\cddefault 1178 \DeclareRobustCommand{\nwwidth}{%
\cdwidth 1179   \not@math@alphabet\nwwidth\relax
\ecdefault 1180   \exfs@merge@width{\nwdefault}}
\ecwidth 1181 \newcommand*{\cddefault}{c}
\ucdefault 1182 \DeclareRobustCommand{\cdwidth}{%
\ucwidth 1183   \not@math@alphabet\cdwidth\relax
1184   \exfs@merge@width{\cddefault}}
1185 \newcommand*{\ecdefault}{ec}
1186 \DeclareRobustCommand{\ecwidth}{%
1187   \not@math@alphabet\ecwidth\relax
1188   \exfs@merge@width{\ecdefault}}
1189 \newcommand*{\ucdefault}{uc}
1190 \DeclareRobustCommand{\ucwidth}{%
1191   \not@math@alphabet\ucwidth\relax
1192   \exfs@merge@width{\ucdefault}}

```

```

\etdefault
\etwidth
\epdefault 1193 \newcommand*{\etdefault}{x}
\epwidth 1194 \DeclareRobustCommand{\etwidth}{%
\exdefault 1195 \not@math@alphabet\etwidth\relax
\exwidth 1196 \exfs@merge@width{\etdefault}}
\uxdefault 1197 \newcommand*{\epdefault}{x}
\uxwidth 1198 \DeclareRobustCommand{\epwidth}{%
1199 \not@math@alphabet\epwidth\relax
1200 \exfs@merge@width{\epdefault}}
1201 \newcommand*{\exdefault}{ex}
1202 \DeclareRobustCommand{\exwidth}{%
1203 \not@math@alphabet\exwidth\relax
1204 \exfs@merge@width{\exdefault}}
1205 \newcommand*{\uxdefault}{ux}
1206 \DeclareRobustCommand{\uxwidth}{%
1207 \not@math@alphabet\uxwidth\relax
1208 \exfs@merge@width{\uxdefault}}

```

`\exfs@merge@weight` :cfr-added merge weight changes into series

```

1209 \newcommand*{\exfs@merge@weight}[1]{%
1210 \edef\exfs@tempg{#1}%
1211 \edef\exfs@tempf{\expandafter\exfs@first\f@series\@nil }%
1212 \edef\exfs@temppart{\expandafter\exfs@part\f@series\@nil }%
1213 \def\templ{1}%
1214 \def\tempb{b}%

```

:case when there's no second part, so the single character must be the weight and should be replaced

```

1215 \ifx\exfs@temppart\@empty
1216 \def\exfs@series{\expandafter\exfs@tempg}%

```

:case when there's a second part

```

1217 \else

```

:get first character of second part

```

1218 \edef\exfs@temps{\expandafter\exfs@second\f@series\@nil }%
1219 \edef\exfs@tempw{\expandafter\exfs@part\exfs@temps\@nil }%

```

:is the first character b? if so, it is part of the weight and should be replaced

```

1220 \ifx\exfs@temps\tempb
1221 \def\exfs@series{\expandafter\exfs@tempg\exfs@tempw}%
1222 \else

```

:is the first character l? if so, it is part of the weight and should be replaced

```

1223 \ifx\exfs@temps\templ
1224 \def\exfs@series{\expandafter\exfs@tempg\exfs@tempw}%
1225 \else

```

:o/w the first character is part of the width and should be retained

```

1226          \def\exfs@series{\expandafter\exfs@tempg\exfs@temppart}%
1227          \fi
1228      \fi
1229  \fi

1230  \ifx\exfs@tempg\exfs@series
1231      \exfs@try@series{\exfs@series}%
1232  \else
1233      \exfs@try@series[\exfs@tempg]{\exfs@series}% assume user wants to
      change weight even if this changes back to the default width
1234  \fi
1235 }
1236 % \end{macrocode}
1237 % \end{macro}
1238 % \begin{macro}{\mbdefault,\mbweight,\bddefault,\bfweight,\bdweight}
1239 % \begin{macrocode}
1240 \newcommand*{\mbdefault}{mb}
1241 \DeclareRobustCommand{\mbweight}{%
1242   \not@math@alphabet\mbweight\relax
1243   \exfs@merge@weight{\mbdefault}}

\dbdefault Heavy weights.
\dbweight
\sbdefault 1244 \newcommand*{\dbdefault}{db}
\sbweight 1245 \DeclareRobustCommand{\dbweight}{%
\sbweight 1246   \not@math@alphabet\dbweight\relax
\ebdefault 1247   \exfs@merge@weight{\dbdefault}}
\ebweight 1248 \newcommand*{\sbdefault}{sb}
\ubdefault 1249 \DeclareRobustCommand{\sbweight}{%
\ubweight 1250   \not@math@alphabet\sbweight\relax
1251   \exfs@merge@weight{\sbdefault}}
1252 \newcommand*{\ebdefault}{eb}
1253 \DeclareRobustCommand{\ebweight}{%
1254   \not@math@alphabet\ebweight\relax
1255   \exfs@merge@weight{\ebdefault}}
1256 \newcommand*{\ubdefault}{ub}
1257 \DeclareRobustCommand{\ubweight}{%
1258   \not@math@alphabet\ubweight\relax
1259   \exfs@merge@weight{\ubdefault}}
1260 \newcommand*{\lgdefault}{l}

\lgdefault note - use this if light is a variant weight, rather than a separate family
\lgweight
\eldefault 1261 \DeclareRobustCommand{\lgweight}{%
\elweight 1262   \not@math@alphabet\lgweight\relax
1263   \exfs@merge@weight{\lgdefault}}
\uldefault 1264 \newcommand*{\eldefault}{el}
\ulweight 1265 \DeclareRobustCommand{\elweight}{%
1266   \not@math@alphabet\elweight\relax
1267   \exfs@merge@weight{\eldefault}}
1268 \newcommand*{\uldefault}{ul}
1269 \DeclareRobustCommand{\ulweight}{%
1270   \not@math@alphabet\ulweight\relax
1271   \exfs@merge@weight{\uldefault}}

```

```

:~end-added

\itshape redefinition
\scshape
1272 \DeclareRobustCommand{\itshape}{%
\upshape
1273 \not@math@alphabet\itshape\mathit
\dfshape
1274 \exfs@merge@shape{\itdefault}{\scdefault}{\sidefault}}

original :cfr-altered: \scshape

1275 \DeclareRobustCommand{\scshape}{%
1276 \not@math@alphabet\scshape\relax
1277 \def\tempu{u}%
1278 \def\tempo{o}%
1279 \ifx\f@shape\tempu
1280 \exfs@merge@shape{\scdefault}{\udefault}{\scudefault}%
1281 \else
1282 \ifx\f@shape\tempo
1283 \exfs@merge@shape{\scdefault}{\oldefault}{\scoldefault}%
1284 \else
1285 \exfs@merge@shape{\scdefault}{\itdefault}{\sidefault}%
1286 \fi
1287 \fi
1288 }

:~end-altered

1289 \DeclareRobustCommand{\upshape}{%
1290 \not@math@alphabet\upshape\relax
1291 \exfs@merge@shape{\updefault}{\sidefault}{\scdefault}}
1292 \DeclareRobustCommand{\dfshape}{%
1293 \not@math@alphabet\dfshape\relax
1294 \fontshape\shapedefault\selectfont}

\swshapedefault
\swshape
1295 \newcommand*{\swshapedefault}{\itdefault}
1296 \DeclareRobustCommand{\swshape}{%
1297 \not@math@alphabet\swshape\relax
1298 \swstyle\fontshape\swshapedefault\selectfont}

\textln
\textos
1299 \DeclareTextFontCommand{\textln}{\lnstyle}
\textin
1300 \DeclareTextFontCommand{\textos}{\osstyle}
\textsu
1301 \DeclareTextFontCommand{\textin}{\instyle}
\textsi
1302 \DeclareTextFontCommand{\textsu}{\sustyle}
\textdf
1303 \DeclareTextFontCommand{\textsi}{\sishape}
\textsw
1304 \DeclareTextFontCommand{\textdf}{\dfshape}
1305 \DeclareTextFontCommand{\textsw}{\swshape}

:cfr-added

\textti Families
\textlt
\textof
\textalt
\textreg
\emboss
\textorn
\textqt
\textsh
\texttm
\texttv

```



```

1306 \DeclareTextFontCommand{\textti}{\tistyle}
1307 \DeclareTextFontCommand{\textlt}{\ltstyle}
1308 \DeclareTextFontCommand{\textof}{\ofstyle}      % open-face (or outline or
        blank) style
1309 \DeclareTextFontCommand{\textalt}{\altstyle}    % alternative style
1310 \DeclareTextFontCommand{\textreg}{\regstyle}    % ‘regular’ style
1311 \DeclareTextFontCommand{\emboss}{\embossstyle}
1312 \DeclareTextFontCommand{\textorn}{\ornamentalstyle} % intended primarily
        for decorative initials etc.
1313 \DeclareTextFontCommand{\textqt}{\qtstyle}
1314 \DeclareTextFontCommand{\textsh}{\shstyle}      % shadowed style
1315 \DeclareTextFontCommand{\texttm}{\tmstyle}
1316 \DeclareTextFontCommand{\texttv}{\tvstyle}

\textl Families - figures
\texto
\textp 1317 \DeclareTextFontCommand{\textl}{\lstyle}
\textt 1318 \DeclareTextFontCommand{\texto}{\ostyle}
\textpl 1319 \DeclareTextFontCommand{\textp}{\pstyle}
\texttp 1320 \DeclareTextFontCommand{\textt}{\tstyle}
\texttll 1321 \DeclareTextFontCommand{\textpl}{\plstyle}
\texttto 1322 \DeclareTextFontCommand{\textpo}{\postyle}
1323 \DeclareTextFontCommand{\texttl}{\tlstyle}
1324 \DeclareTextFontCommand{\textto}{\tostyle}

\textol Shapes
\textswash
\textu 1325 \DeclareTextFontCommand{\textol}{\olshape}      % outline
1326 \DeclareTextFontCommand{\textswash}{\swashstyle} % an attempt to improve
\textscu on \textsw
\textui 1327 \DeclareTextFontCommand{\textu}{\ushape}
\texttri 1328 \DeclareTextFontCommand{\textscu}{\scushape}
1329 \DeclareTextFontCommand{\textui}{\uishape}      % upright italic
1330 \DeclareTextFontCommand{\texttri}{\rishape}      % reverse italic

\textnw Widths
\textcd
\textec 1331 \DeclareTextFontCommand{\textnw}{\nwwidth}
\textuc 1332 \DeclareTextFontCommand{\textcd}{\cdwidth}
\textet 1333 \DeclareTextFontCommand{\textec}{\ecwidth}
1334 \DeclareTextFontCommand{\textuc}{\ucwidth}
\textep 1335 \DeclareTextFontCommand{\textet}{\etwidth}
\textex 1336 \DeclareTextFontCommand{\textep}{\epwidth}
\textux 1337 \DeclareTextFontCommand{\textex}{\exwidth}
\textrw 1338 \DeclareTextFontCommand{\textux}{\uxwidth}
1339 \DeclareTextFontCommand{\textrw}{\regwidth}

\textmb Weights
\textdb 1340 \DeclareTextFontCommand{\textmb}{\mbweight}
\textbd 1341 \DeclareTextFontCommand{\textdb}{\dbweight}
\textsb 1342 \DeclareTextFontCommand{\textsb}{\sbweight}
\texteb 1343 \DeclareTextFontCommand{\texteb}{\ebweight}
\textub 1344 \DeclareTextFontCommand{\textub}{\ubweight}
\textlg
\textel
\textul

```

```

1345 \DeclareTextFontCommand{\textlg}{\lgweight}
1346 \DeclareTextFontCommand{\textel}{\elweight}
1347 \DeclareTextFontCommand{\textul}{\ulweight}

```

end-added

Change History

SVN6140	General: Fixes a bug which prevented <code>\tmstyle</code> and <code>\tvstyle</code> working correctly if the current font was not a serif family. (Especially problematic in Beamer where <code>\normalfont</code> cannot be used as a workaround, but annoying elsewhere.) 1	force : Add option force . Load old file/incompatible with force . . 13
2008-10-26	General: First public release as part of cfr-lm 1	nfssect-cfr : Behaviour depends on kernel version and options. On newer kernels, quite conservative/less compatible by default. Fully compatible on older kernels. 12
2008-12-22	General: Updated version released standalone. 1	So nfssect-cfr merging is now limited to family, weight and width. 12
2010-07-17	General: 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. 12	Split nfssect-cfr.sty into nfssect-cfr{,-nfss,-nnfss}.sty 12
	\Microtype@Hook : Add microtype support for variants. 15	nfssect-cfr-nnfss : Conditionally override kernel rules affecting switches to upright/small-caps.italic etc. . 28
	\qtstyle : Improve \ofstyle . . . 25	Unconditionally add a bunch of shape change rules for shapes unsupported by the kernel - I don't *think* these should be problematic: if the current or requested shape is unsupported by the kernel, surely it can't be problematic to support that shape? 28
v0.0	\nfssect-cfr : Update for NNFSS. . 12	\swshape : Conditionally overwrite \swshape to take account of default setting. This is not for any package I know of on CTAN, but the original code used \textsw 33
v1.0	compat : Add option compat . More aggressive/backwards compatible with compat 13	\ubweight : Make \mbdefault sb (duplicating \sbdefault) as I can't come up with anything better. 32
	debug : Add option debug 13	
	\exfs@merge@width : Do not depend on incorrect series names, which are no longer supported. 30	v6140
	\exfs@series@splitter : Rewritten as kernel no longer supports erroneous m 21	General: Extend documentation somewhat. 12
		\tvstyle : Modify \tmstyle and \tvstyle to unmerge sans and other typewriter before merging appropriate variant. 26

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.

Symbols

`\/` 112
`\@empty` .. 198, 229, 336, 337, 338,
 345, 346, 378, 381, 388, 389,
 404, 416, 427, 432, 444, 504,
 507, 519, 608, 609, 657, 659,
 843, 864, 908, 912, 919, 920,
 937, 949, 960, 965, 977, 1037,
 1040, 1052, 1114, 1136, 1158, 1215
`\@for` 387, 415, 918, 948
`\@ifl@t@r` 16, 41
`\@ifpackageloaded` 138, 738
`\@ifundefined` 7, 145
`\@nfssectcfr@digonnewfalse` ... 12, 19
`\@nfssectcfr@digonnewtrue` 6
`\@nil` 113, 120, 131, 290,
 293, 299, 304, 322, 323, 324,
 325, 326, 327, 328, 330, 331,
 344, 351, 357, 358, 361, 364,
 368, 371, 375, 391, 394, 397,
 409, 423, 436, 447, 473, 503,
 511, 513, 518, 527, 880, 881,
 882, 883, 884, 885, 886, 889,
 890, 894, 897, 900, 903, 906,
 922, 927, 930, 942, 956, 969,
 980, 1006, 1036, 1044, 1046,
 1051, 1059, 1112, 1113, 1117, 1119,
 1122, 1132, 1133, 1139, 1154,
 1155, 1161, 1211, 1212, 1218, 1219
`\@roman` 110, 122, 129
`__exfs_set:n` 103, 107

A

`\addedfalse` 911
`\addedtrue` ... 914, 938, 953, 959, 964
`\altstyle` 5, 459, 762, 995, 1309

B

`\bddefault` 670, 1238
`\bdweight` 7, 670, 795, 1238
`\begin` 1238, 1239
`\begingroup` 111,
 128, 193, 221, 256, 280, 839, 860
`\bfweight` 7, 670, 1238
`\bool_if:NT` 93, 555
`\bool_if:NTF` 88, 718

C

`\catcode` 112
`\cddefault` 616, 1177
`\cdwidth` 7, 616, 785, 1177, 1332
`\char` 132
`compat (opt.)` 8, 29
`\cs_if_exist:NTF` 86
`\cs_if_exist_use:c` 69
`\cs_new_protected_nopar:Nn` 103
`\cs_set_eq:NN` 107
`\csname` 110, 122, 129,
 159, 173, 196, 244, 841, 862, 1105
`\curr@fontshape`
 .. 159, 170, 171, 174, 176, 178,
 180, 186, 188, 196, 275, 841, 862

D

`\dbdefault` 685, 1244
`\dbweight` 7, 685, 794, 1244, 1341
`debug (opt.)` 11, 33
`\DeclareFontShapeChangeRule` 544,
 545, 546, 547, 548, 549, 550,
 551, 552, 553, 557, 558, 559,
 560, 561, 564, 565, 566, 567,
 568, 569, 570, 571, 572, 573,
 574, 575, 576, 577, 578, 579,
 580, 581, 582, 583, 584, 585, 586
`\DeclareMicrotypeVariants` 136
`\DeclareRobustCommand` .. 121, 355,
 359, 362, 366, 369, 450, 453,
 456, 459, 462, 465, 468, 471,
 474, 477, 480, 484, 487, 490,
 495, 529, 532, 535, 537, 539,
 542, 588, 591, 594, 597, 600,
 603, 613, 617, 621, 625, 629,
 633, 637, 641, 645, 649, 671,
 675, 678, 682, 686, 690, 694,
 698, 702, 706, 710, 720, 729,
 887, 892, 895, 898, 901, 983,
 986, 989, 992, 995, 998, 1001,
 1004, 1007, 1010, 1013, 1017,
 1020, 1023, 1028, 1061, 1064,
 1067, 1069, 1071, 1074, 1078,
 1082, 1086, 1090, 1094, 1098,
 1174, 1178, 1182, 1186, 1190, 1194,
 1198, 1202, 1206, 1241, 1245,

- 1249, 1253, 1257, 1261, 1265,
1269, 1272, 1275, 1289, 1292, 1296
- `\DeclareTextFontCommand`
 749, 753, 754, 755, 756,
 757, 758, 759, 760, 761, 762,
 763, 764, 765, 766, 767, 768,
 769, 770, 771, 772, 773, 774,
 775, 776, 777, 778, 779, 780,
 781, 782, 783, 784, 785, 786,
 787, 788, 789, 790, 791, 792,
 793, 794, 795, 796, 797, 798,
 799, 800, 801, 1299, 1300, 1301,
 1302, 1303, 1304, 1305, 1306,
 1307, 1308, 1309, 1310, 1311,
 1312, 1313, 1314, 1315, 1316,
 1317, 1318, 1319, 1320, 1321,
 1322, 1323, 1324, 1325, 1326,
 1327, 1328, 1329, 1330, 1331,
 1332, 1333, 1334, 1335, 1336,
 1337, 1338, 1339, 1340, 1341,
 1342, 1343, 1344, 1345, 1346, 1347
- `\DeclareTextOrnament` 109
- `\def` 110, 114, 115,
 116, 117, 118, 120, 135, 243, 322,
 323, 324, 325, 326, 327, 328,
 329, 337, 338, 343, 379, 655,
 880, 881, 882, 883, 884, 885,
 886, 909, 1115, 1134, 1135, 1137,
 1141, 1144, 1146, 1156, 1157, 1159,
 1163, 1166, 1168, 1213, 1214,
 1216, 1221, 1224, 1226, 1277, 1278
- `\DefineFileInfoSVN` 4, 151, 832
- `\dfshape` 6, 713, 758, 1272, 1304
- `\do` 387, 415, 918, 948
- E**
- `\ebdefault` 685, 1244
- `\ebweight` 7, 685, 797, 1244, 1343
- `\ecdefault` 616, 1177
- `\ecwidth` 7, 616, 786, 1177, 1333
- `\edef` 129, 197, 216, 223, 228,
 244, 249, 257, 258, 330, 331,
 344, 345, 346, 347, 350, 374,
 375, 377, 382, 391, 394, 397,
 400, 403, 408, 409, 419, 423,
 425, 430, 435, 436, 502, 503,
 505, 508, 511, 513, 514, 517, 518,
 520, 607, 610, 662, 842, 863,
 905, 906, 907, 913, 922, 924,
 925, 927, 930, 933, 936, 941,
 942, 952, 956, 958, 963, 968,
 969, 1035, 1036, 1038, 1041,
 1044, 1046, 1047, 1050, 1051,
 1053, 1102, 1103, 1107, 1112,
 1113, 1117, 1119, 1122, 1124, 1131,
 1132, 1133, 1139, 1154, 1155,
 1161, 1210, 1211, 1212, 1218, 1219
- `\eldefault` 701, 1261
- `\else` 127, 165,
 187, 202, 208, 220, 227, 233,
 248, 262, 265, 273, 296, 302,
 307, 312, 317, 333, 334, 335,
 336, 338, 346, 347, 384, 390,
 401, 406, 414, 418, 422, 428,
 433, 446, 509, 516, 608, 609,
 658, 666, 847, 853, 868, 874,
 915, 921, 934, 939, 947, 951,
 955, 961, 966, 979, 1042, 1049,
 1106, 1116, 1120, 1123, 1138,
 1142, 1145, 1160, 1164, 1167,
 1217, 1222, 1225, 1232, 1281, 1284
- `\elweight` 7, 701, 800, 1261, 1346
- `\emboss` 5, 759, 1306
- `\embossstyle` ... 5, 459, 764, 995, 1311
- `\end` 1236, 1237
- `\endcsname` 110, 122, 129,
 159, 170, 173, 174, 176, 178, 196,
 224, 244, 259, 260, 841, 862, 1105
- `\endgroup` 119,
 133, 212, 240, 268, 320, 856, 877
- `\epdefault` 632, 1193
- `\epwidth` 7, 632, 789, 1193, 1336
- `\etdefault` 632, 1193
- `\etwidth` 7, 632, 788, 1193, 1335
- `\exdefault` 632, 1193
- `\exf@try@family` 290
- `\exfs@addedfalse` 380
- `\exfs@addedtrue` 383, 405, 420, 426, 431
- `\exfs@base@family` .. 120, 122, 124, 129
- `\exfs@check@cx` 332, 333, 334, 335, 343
- `\exfs@first` 322,
 330, 344, 880, 1112, 1132, 1154, 1211
- `\exfs@font@width` 1111
- `\exfs@get@base` 120,
 290, 293, 299, 304, 322, 357,
 358, 361, 364, 368, 371, 447,
 473, 527, 880, 889, 890, 894,
 897, 900, 903, 980, 1006, 1059
- `\exfs@get@variants`
 .. 322, 375, 503, 880, 906, 1036
- `\exfs@info` .. 162, 171, 186, 188, 194,
 203, 209, 218, 225, 246, 250,
 275, 341, 376, 611, 664, 667, 741
- `\exfs@merge@families` 282,
 285, 288, 309, 314, 318, 372,

452, 455, 458, 461, 464, 467,
 470, 476, 479, 483, 486, 489,
 494, 499, 904, 985, 988, 991,
 994, 997, 1000, 1003, 1009,
 1012, 1016, 1019, 1022, 1027, 1032
 \exfs@merge@shape
 1101, 1274, 1280, 1283, 1285, 1291
 \exfs@merge@weight 652, 673,
 677, 680, 684, 688, 692, 696,
 700, 704, 708, 712, 1209, 1247,
 1251, 1255, 1259, 1263, 1267, 1271
 \exfs@merge@width 605, 615,
 619, 623, 627, 631, 635, 639,
 643, 647, 651, 1130, 1180, 1184,
 1188, 1192, 1196, 1200, 1204, 1208
 \exfs@next 322,
 391, 423, 511, 880, 922, 956, 1044
 \exfs@nextvariant 389,
 391, 393, 396, 407, 423, 424, 434
 \exfs@normalise 169, 222
 \exfs@part 322, 880, 1113,
 1119, 1122, 1133, 1155, 1212, 1219
 \exfs@reserved 228, 229
 \exfs@second
 .. 322, 880, 1117, 1139, 1161, 1218
 \exfs@series ... 610, 611, 612, 662,
 663, 664, 665, 667, 668, 1137,
 1141, 1144, 1146, 1150, 1152, 1159,
 1163, 1166, 1168, 1172, 1216,
 1221, 1224, 1226, 1230, 1231, 1233
 \exfs@series@splitter . 329, 606, 653
 \exfs@shift 322, 331, 351,
 394, 397, 409, 436, 513, 518,
 880, 927, 930, 942, 969, 1046, 1051
 \exfs@split@orndef 113, 130
 \exfs@swfamily 261, 266, 278
 \exfs@swshape 254, 732
 \exfs@takefalse 515, 521
 \exfs@taketrue 506
 \exfs@targetseries 215, 226, 241
 \exfs@targetshape
 ... 244, 245, 249, 250, 251, 252
 \exfs@targetsw 255, 263, 269,
 274, 279, 297, 303, 308, 313, 318
 \exfs@tempa 118, 132,
 152, 192, 197, 198, 210, 213, 216,
 217, 223, 224, 257, 258, 259,
 260, 281, 283, 286, 289, 291,
 294, 344, 345, 346, 348, 834,
 838, 842, 843, 854, 857, 859,
 863, 864, 875, 878, 1102, 1107, 1110
 \exfs@tempa@fake 258
 \exfs@tempb 129, 131, 152, 834, 1103, 1104
 \exfs@tempf
 . 152, 502, 512, 834, 905, 913,
 917, 952, 957, 962, 1035, 1045,
 1112, 1132, 1137, 1141, 1144, 1146,
 1154, 1159, 1163, 1166, 1168, 1211
 \exfs@tempg 379, 400, 403,
 408, 419, 425, 430, 435, 505,
 514, 517, 520, 654, 909, 933,
 936, 941, 952, 958, 963, 968,
 1038, 1047, 1050, 1053, 1210,
 1216, 1221, 1224, 1226, 1230, 1233
 \exfs@temph 607,
 608, 609, 610, 655, 656, 659,
 662, 663, 1131, 1137, 1141, 1144, 1146
 \exfs@tempn 511, 512,
 517, 920, 922, 926, 929, 940,
 956, 957, 967, 1044, 1045, 1050
 \exfs@temppart 1113, 1114, 1124, 1133,
 1136, 1155, 1158, 1212, 1215, 1226
 \exfs@tempq 378, 382,
 403, 419, 425, 430, 444, 447,
 504, 508, 514, 520, 527, 908,
 913, 936, 952, 958, 963, 977,
 980, 1037, 1041, 1047, 1053, 1059
 \exfs@temps 1117,
 1118, 1119, 1121, 1122, 1139, 1140,
 1141, 1143, 1144, 1161, 1162, 1163,
 1165, 1166, 1218, 1219, 1220, 1223
 \exfs@tempw 1219, 1221, 1224
 \exfs@try@family 191,
 292, 357, 361, 364, 368, 371,
 447, 473, 527, 837, 889, 894,
 897, 900, 903, 980, 1006, 1059
 \exfs@try@series 214, 612,
 619, 623, 627, 635, 639, 665,
 668, 692, 696, 700, 704, 708,
 712, 858, 1150, 1172, 1231, 1233
 \exfs@try@shapeshift 243,
 543, 589, 592, 595, 598, 601, 604
 \exfs@unmerge@families . 492, 493,
 497, 498, 501, 531, 534, 1025,
 1026, 1030, 1031, 1033, 1063, 1066
 \exfs@unmerge@width 1153, 1176
 \exfs@unmerge@families 500
 \exfs@variants 375, 376, 381, 388,
 391, 394, 397, 403, 404, 409,
 416, 423, 425, 427, 430, 432, 436
 \exfs@vartomerge
 374, 382, 386, 419, 424, 429
 \exfs@weight 330, 332, 333,
 334, 335, 337, 338, 345, 346,
 348, 349, 608, 609, 610, 667, 668
 \exfs@weight: 341

- \exfs@weighta 347, 349
 - \exfs@width 331, 336, 337, 338, 341,
344, 345, 346, 351, 352, 657,
662, 1115, 1119, 1122, 1124, 1128
 - \exfs@widtha 350, 352
 - \expandafter 110, 120, 122, 130, 131,
158, 159, 173, 196, 290, 293,
298, 303, 330, 331, 344, 351,
357, 358, 361, 364, 368, 371,
375, 391, 394, 397, 409, 423,
436, 447, 473, 503, 511, 513,
518, 527, 841, 862, 889, 890,
894, 897, 900, 903, 906, 922,
927, 930, 942, 956, 969, 980,
1006, 1036, 1044, 1046, 1051,
1059, 1105, 1112, 1113, 1117, 1119,
1122, 1132, 1133, 1137, 1139,
1141, 1144, 1146, 1154, 1155, 1159,
1161, 1163, 1166, 1168, 1211, 1212,
1216, 1218, 1219, 1221, 1224, 1226
 - \ExpandArgs 68
 - \ExplLoaderFileDate 16
 - \ExplSyntaxOff 108, 161, 563, 752
 - \ExplSyntaxOn 25, 155, 554, 715
 - \exwidth 7, 632, 790, 1193, 1337
- F**
- \f@encoding
. 114, 194, 200, 204, 205, 209,
223, 231, 235, 236, 251, 257,
845, 849, 850, 866, 870, 871, 1105
 - \f@family 115, 120,
223, 231, 235, 236, 251, 257,
281, 283, 286, 289, 290, 291,
293, 294, 299, 304, 357, 358,
361, 364, 368, 371, 375, 447,
473, 503, 527, 866, 870, 871,
889, 890, 894, 897, 900, 903,
906, 980, 1006, 1036, 1059, 1105
 - \f@series .. 116, 175, 217, 218, 225,
251, 257, 606, 653, 654, 1105,
1112, 1113, 1117, 1132, 1133, 1139,
1154, 1155, 1161, 1211, 1212, 1218
 - \f@shape 117, 175, 223, 231, 235, 236,
245, 246, 250, 1104, 1279, 1282
 - \fi 21, 134, 167, 183, 184, 185, 189,
207, 211, 238, 239, 242, 253,
264, 267, 276, 301, 306, 311,
316, 319, 339, 340, 353, 392,
395, 398, 410, 411, 412, 421, 437,
438, 439, 440, 442, 443, 448,
522, 523, 524, 526, 608, 609,
660, 661, 669, 852, 855, 873,
876, 923, 928, 931, 943, 944,
945, 954, 970, 971, 972, 973,
975, 976, 981, 1055, 1056, 1058,
1108, 1109, 1125, 1126, 1127,
1147, 1148, 1149, 1169, 1170, 1171,
1227, 1228, 1229, 1234, 1286, 1287
- G**
- \g@addto@macro 147
 - \g__exfs_compat_bool 29, 93, 555, 718
 - \g__exfs_force_bool 36, 88
 - \g_msg_module_name_prop 26
 - \gdef . 113, 210, 226, 263, 297, 303,
308, 313, 318, 385, 854, 875, 916
 - \global 146, 158
- H**
- \hook_gput_code:nnn . 84, 156, 716, 739
- I**
- \if .. 286, 289, 291, 294, 332, 333,
334, 335, 337, 338, 345, 346,
393, 396, 399, 402, 608, 609, 656
 - \if@nfssectcfr@digonnew ... 5, 15, 23
 - \ifadded 910, 950
 - \ifcsname 170, 174, 176, 178, 224, 259, 260
 - \ifexfs@added 372
 - \ifexfs@debug 24, 163
 - \ifexfs@take 500
 - \IfFileExists 8
 - \IfFormatAtLeastTF 41, 42, 65
 - \IfValueTF 50
 - \ifx 122, 196, 198,
217, 229, 245, 269, 283, 336,
381, 386, 388, 407, 416, 424,
429, 434, 444, 507, 512, 519,
608, 609, 657, 663, 841, 843,
862, 864, 912, 917, 919, 926,
929, 932, 935, 940, 949, 957,
962, 967, 977, 1040, 1045, 1052,
1104, 1105, 1114, 1118, 1121, 1136,
1140, 1143, 1158, 1162, 1165,
1215, 1220, 1223, 1230, 1279, 1282
 - \infstyle 6, 355, 749, 755, 887
 - \init@series@setup . 86, 802, 811, 820
 - \instyle 6, 355, 887, 1301

- `\itdefault` 714, 1274, 1285, 1295
`\itshape` 4, 1272
- K**
- `\keys_define:nn` 27
`\keys_set:nn` 105
- L**
- `\let` 146, 158,
 192, 215, 255, 279, 281, 337,
 338, 345, 346, 349, 352, 365,
 378, 389, 404, 427, 432, 504,
 608, 609, 654, 659, 713, 838,
 859, 908, 920, 937, 960, 965, 1037
`\lgdefault` 701, 1260, 1261
`\lgweight` 7, 701, 799, 1261, 1345
`\lnstyle` 6, 355, 753, 887, 1299
`\loop` 510
`\lstyle` 6, 529, 536,
 538, 770, 1061, 1068, 1070, 1317
`\ltstyle` 5, 459, 760, 995, 1307
- M**
- `\mathit` 1273
`\mbdefault` 670, 1238, 1240, 1243
`\mbweight` 7,
 670, 793, 1238, 1241, 1242, 1340
`\mdwdefault` 648
`\mdweight` 7, 681
`\mdwidth` 7, 648
`\MessageBreak` 52, 56, 58, 124,
 140, 141, 171, 179, 181, 200, 204,
 218, 231, 235, 246, 250, 271,
 742, 743, 744, 845, 849, 866, 870
`\Microtype@Hook` 135
`\msg_line_number:` 74, 80
`\msg_new:nnn` 72, 78
`\msg_warning:nn` 90, 95
`\mwdefault` 681
- N**
- `\newcommand` 109,
 152, 153, 154, 162, 169, 191, 214,
 254, 278, 373, 501, 541, 587,
 590, 593, 596, 599, 602, 605,
 616, 620, 624, 628, 632, 636,
 640, 644, 648, 652, 670, 674,
 681, 685, 689, 693, 697, 701,
 705, 709, 714, 834, 835, 836,
 837, 858, 904, 1034, 1073, 1077,
 1081, 1085, 1089, 1093, 1097,
 1101, 1111, 1130, 1153, 1177, 1181,
 1185, 1189, 1193, 1197, 1201,
 1205, 1209, 1240, 1244, 1248,
 1252, 1256, 1260, 1264, 1268, 1295
`\newcounter` 1033
`\NewDocumentCommand` 48
`\newif` 5, 24, 372, 500, 910
`nfssect-cfr` (pkg.) 1
`nfssect-cfr-nfss` (pkg.) 829
`nfssect-cfr-nnfss` (pkg.) 148
`\nfssectcfr@MT@Hook` 135
`\nfssectset` 4, 107
`\normalfont` 177
`\normalshape` 713
`\not@math@alphabet` 249,
 356, 360, 363, 367, 370, 451,
 454, 457, 460, 463, 466, 469,
 472, 475, 478, 481, 485, 488,
 491, 496, 530, 533, 614, 618,
 622, 626, 630, 634, 638, 642,
 646, 650, 672, 676, 679, 683,
 687, 691, 695, 699, 703, 707,
 711, 722, 731, 888, 893, 896,
 899, 902, 984, 987, 990, 993,
 996, 999, 1002, 1005, 1008,
 1011, 1014, 1018, 1021, 1024,
 1029, 1062, 1065, 1075, 1079,
 1083, 1087, 1091, 1095, 1099,
 1175, 1179, 1183, 1187, 1191, 1195,
 1199, 1203, 1207, 1242, 1246,
 1250, 1254, 1258, 1262, 1266,
 1270, 1273, 1276, 1290, 1293, 1297
`\nwdefault` 616, 1177
`\nwwidth` 7, 616, 784, 1177, 1331
- O**
- `\ofstyle` 5, 459, 761, 995, 1308
`\oldefault` 587, 1077, 1283
`\olshape` 6, 587, 778, 1077, 1325
 options:
 `compat` 8, 29
 `debug` 11, 33
 `force` 8, 36
`\ornament` 5, 109
`\ornamentalstyle` 5, 459, 765, 995, 1312
`\osstyle` 6, 355, 754, 887, 1300
`\ostyle` 6, 450, 540, 771, 983, 1072, 1318
- P**
- `\PackageError` 51, 179, 445, 978
`\PackageInfo` 166, 848, 869
`\PackageWarning` 123, 139, 164, 199,
 230, 234, 270, 725, 734, 803,
 807, 812, 816, 821, 825, 844, 865

- \patchcmd 802, 811, 820
- \plstyle 6, 535, 774, 1067, 1321
- \postyle 6, 450, 775, 983, 1322
- \ProcessKeyOptions 48
- \ProcessKeysOptions 47, 50, 58
- \prop_gput:Nnn 26
- \protect 55, 56, 58, 60, 742, 745
- \providecommand 41, 68
- \pstyle 6, 450, 538, 772, 983, 1070, 1319

- Q**
- \qtstyle 5, 459, 766, 995, 1313

- R**
- \regstyle 5, 459, 763, 995, 1310
- \regwidth 7, 613, 792, 1174, 1339
- \relax . 122, 159, 192, 196, 215, 217,
245, 249, 255, 269, 279, 295,
356, 360, 363, 367, 370, 451,
454, 457, 460, 463, 466, 469,
472, 475, 478, 481, 485, 488,
491, 496, 530, 533, 614, 618,
622, 626, 630, 634, 638, 642,
646, 650, 656, 657, 672, 676,
679, 683, 687, 691, 695, 699,
703, 707, 711, 722, 731, 838,
841, 859, 862, 888, 893, 896,
899, 902, 984, 987, 990, 993,
996, 999, 1002, 1005, 1008,
1011, 1014, 1018, 1021, 1024,
1029, 1062, 1065, 1075, 1079,
1083, 1087, 1091, 1095, 1099,
1105, 1175, 1179, 1183, 1187, 1191,
1195, 1199, 1203, 1207, 1242,
1246, 1250, 1254, 1258, 1262,
1266, 1270, 1276, 1290, 1293, 1297
- \repeat 524, 525
- \RequirePackage .. 2, 9, 11, 18, 46,
67, 71, 91, 97, 100, 149, 830, 833
- \revinfo 3, 150, 831
- \ridefault 599, 1093
- \rishape 6, 599, 783, 1093, 1330
- \rule 126

- S**
- \sbdefault 685, 1244
- \sbweight 7, 685, 796, 1244, 1342
- \scdefault 1274, 1280, 1283, 1285, 1291
- \scoldefault 587, 1077, 1283
- \scolshape 6, 587, 1077
- \scshape 4, 576, 582, 1272
- \scudefault 593, 1085, 1280
- \scushape 6, 593, 781, 1085, 1328
- \selectfont
. 132, 173, 175, 210, 226, 252,
263, 299, 304, 723, 854, 875,
1076, 1080, 1084, 1088, 1092,
1096, 1100, 1110, 1152, 1294, 1298
- \seriesdefault 175
- \set 385, 387, 415, 916, 918, 948
- \setcounter 1039
- \sffamily 482, 1015
- \shapedefault 175, 1294
- \shstyle 5, 459, 767, 995, 1314
- \sidefault . 541, 1073, 1274, 1285, 1291
- \sishape 6, 541, 757, 1073, 1303
- \space 225, 250, 726, 735
- \stepcounter 1048, 1054
- \string 131
- \sustyle 6, 355, 756, 887, 1302
- \swashapedefault 714, 1295
- \swashstyle 5, 459, 779, 995, 1326
- \swdefault 257, 263
- \swshape .. 4, 5, 6, 717, 718, 1295, 1305
- \swshapedefault 287,
292, 298, 309, 714, 723, 1295, 1298
- \swstyle 6, 355, 487, 723, 887, 1020, 1298

- T**
- \tempa . 503, 507, 511, 513, 514, 518,
519, 906, 912, 919, 922, 927,
930, 936, 937, 942, 949, 956,
958, 960, 963, 965, 969, 1036,
1040, 1044, 1046, 1047, 1051, 1052
- \tempb 1134, 1140, 1156, 1162, 1214, 1220
- \tempj 925, 929, 935
- \templ 1135, 1143, 1157, 1165, 1213, 1223
- \tempo .. 377, 386, 907, 917, 1278, 1282
- \tempt 924, 926, 932
- \tempu 1277, 1279
- \textalt 5, 759, 1306
- \textbackslash 180
- \textbd 7, 793, 1340
- \textcd 7, 784, 1331
- \textdb 7, 793, 1340
- \textdf 6, 753, 1299
- \texteb 7, 793, 1340
- \textec 7, 784, 1331
- \textel 7, 793, 1340
- \textep 7, 784, 1331
- \textet 7, 784, 1331
- \textex 7, 784, 1331
- \textin 6, 738, 1299
- \textinf 6, 745, 753

<code>\textl</code>	6 , 770 , 1317	<code>\textul</code>	7 , 793 , 1340
<code>\textlg</code>	7 , 793 , 1340	<code>\textux</code>	7 , 784 , 1331
<code>\textln</code>	6 , 753 , 1299	<code>\tistyle</code>	5 , 459 , 759 , 983 , 1306
<code>\textlt</code>	5 , 759 , 1306	<code>\tlstyle</code>	6 , 535 , 776 , 1067 , 1323
<code>\textmb</code>	7 , 793 , 1340	<code>\tmstyle</code>	5 , 490 , 768 , 1023 , 1315
<code>\textnw</code>	7 , 784 , 1331	<code>\tostyle</code>	6 , 535 , 777 , 1067 , 1324
<code>\texto</code>	6 , 770 , 1317	<code>\try@load@fontshape</code> ...	195 , 840 , 861
<code>\textof</code>	5 , 759 , 1306	<code>\tstyle</code>	6 , 529 , 536 , 540 , 773 , 1061 , 1068 , 1072 , 1320
<code>\textol</code>	6 , 778 , 1325	<code>\tvstyle</code>	5 , 490 , 769 , 1023 , 1316
<code>\textorn</code>	5 , 759 , 1306	U	
<code>\textos</code>	6 , 753 , 1299	<code>\ubdefault</code>	685 , 1244
<code>\textp</code>	6 , 770 , 1317	<code>\ubweight</code>	7 , 685 , 798 , 1244 , 1344
<code>\textpl</code>	6 , 770 , 1317	<code>\ucdefault</code>	616 , 1177
<code>\textpo</code>	6 , 770 , 1317	<code>\ucwidth</code>	7 , 616 , 787 , 1177 , 1334
<code>\textqt</code>	5 , 759 , 1306	<code>\udefault</code>	593 , 1085 , 1280
<code>\textreg</code>	5 , 759 , 1306	<code>\uidefault</code>	599 , 1093
<code>\textri</code>	6 , 778 , 1325	<code>\uishape</code>	6 , 599 , 782 , 1093 , 1329
<code>\textrw</code>	7 , 784 , 1331	<code>\uldefault</code>	701 , 1261
<code>\textsb</code>	7 , 793 , 1340	<code>\ulweight</code>	7 , 701 , 801 , 1261 , 1347
<code>\textscu</code>	6 , 778 , 1325	<code>\updefault</code>	1291
<code>\textsh</code>	5 , 759 , 1306	<code>\upshape</code>	4 , 1272
<code>\textsi</code>	6 , 753 , 1299	<code>\ushape</code>	6 , 593 , 780 , 1085 , 1327
<code>\textsu</code>	6 , 753 , 1299	<code>\uxdefault</code>	632 , 1193
<code>\textsw</code>	5 , 6 , 779 , 1299 , 1326	<code>\uxwidth</code>	7 , 632 , 791 , 1193 , 1338
<code>\textswash</code>	5 , 778 , 1325	V	
<code>\texttt</code>	6 , 770 , 1317	<code>\value</code>	1043
<code>\texttti</code>	5 , 759 , 1306	W	
<code>\textttl</code>	6 , 770 , 1317	<code>\whiledo</code>	1043
<code>\texttm</code>	5 , 759 , 1306	X	
<code>\textto</code>	6 , 770 , 1317	<code>\xx</code> 399 , 400 , 402 , 403 , 407 , 408 , 429 ,	
<code>\texttv</code>	5 , 759 , 1306	430 , 434 , 435 , 932 , 933 , 935 ,	
<code>\texttu</code>	6 , 778 , 1325	936 , 940 , 941 , 962 , 963 , 967 , 968	
<code>\textub</code>	7 , 793 , 1340	<code>\xx:</code>	387 , 415 , 918 , 948
<code>\textuc</code>	7 , 784 , 1331		
<code>\textui</code>	6 , 778 , 1325		