

# MinionPro Support for L<sup>A</sup>T<sub>E</sub>X

Achim Blumensath

Andreas Böhmann

Sebastian Schubert

Michael Zedler

v2.2 – 2011/12/19

## Contents

1	Overview	2
2	Interference with other packages	2
3	Options	2
4	Figure selection	4
5	Additional font shapes and symbols	5
6	Language support	6
7	Searching for figures or for words containing ligatures in PDF documents	6
8	NFSS classification	7
9	Version history	7
10	The main style file	8
10.1	Options . . . . .	8
10.2	Font declarations . . . . .	12
10.3	Font selection . . . . .	13
10.4	Greek letters . . . . .	14
10.5	pdfT <sub>E</sub> X to-unicode support . . . . .	16
10.6	Superior and inferior figures . . . . .	18
10.7	Additional symbols . . . . .	20
10.8	Integral symbols . . . . .	21
10.9	Open G support . . . . .	23
10.10	Logos . . . . .	23
10.11	AMS . . . . .	23
11	Support for character protrusion	24

## 1 Overview

The MinionPro package provides support for the MinionPro font family from Adobe. You can use these fonts in a  $\text{\LaTeX}$  document by adding the command

```
\usepackage{MinionPro}
```

to the preamble. This will change both the text font and the math font to MinionPro. If you prefer another math font (such as `eulervm`) use the option `onlytext` as explained in Section 3.

## 2 Interference with other packages

The MinionPro package automatically loads the following packages: `textcomp`, `amsmath`, `fontaxes` and `MnSymbol` (version 1.4). If you want to pass options to these packages you can either put the corresponding `\usepackage` command before the `\usepackage{MinionPro}` or you can include the options in the `\documentclass` command. The MinionPro package is *not* compatible with `amssymb` and `amsfonts`. Please see also the corresponding section in the `MnSymbol` documentation.

The MinionPro package includes support files for the microtype package (version 1.8 or higher), consult the package's documentation for further details.

There is also a slight incompatibility with the `dcolum` package which expects all figures to have the same width. If you want to use this package you either have to specify the `mathtabular` option (this is the brute force solution, not recommended), or you can use the `\figureversion{tabular}` command to switch to tabular figures in front of every table (much better, but also more work). In addition, `dcolum` sets figures in math mode, hence the choice of math figures (see Section 3) determines if text or lining figures are used.

## 3 Options

### Font selection

The following options specify which version of the fonts you want to use. The default settings are marked with an asterisk\*.

<code>smallfamily*</code>	use only regular and bold face
<code>medfamily</code>	use semibold face in addition to <code>smallfamily</code>
<code>fullfamily</code>	use medium face in addition to <code>medfamily</code>
<code>noopticals*</code>	use only the optical size Text
<code>opticals</code>	use the optical sizes Caption, Text, Subhead, and Display
<code>slides</code>	use only the optical size Caption (useful for slides)
<code>normalsize*</code>	adapt optical sizes to the normal font size (10 pt, 11 pt, 12 pt)

<code>nonnormalsize</code>	use static settings for the optical sizes
----------------------------	---

Since MinionPro comes in only four different optical sizes we use a variable mapping from font size to the optical size. This means that, both for 10 pt and 11 pt documents, text set in `\small` size will use the Caption size. Sometimes it might be desirable to turn off this automatism – for instance, if you want to load the MinionPro package before the `\documentclass` command. In these cases you can use the `nonnormalsize` option to do so.

The package also provides a way to only change the text fonts or only the math fonts.

<code>onlytext</code>	only change the text fonts
<code>onlymath</code>	only change the math fonts

### Figure selection

MinionPro offers four different figure versions. A detailed description is given in Section 4. The default version can be selected by the following options:

<code>textosf</code>	use text figures in text mode
<code>mathosf</code>	use text figures in math mode
<code>osf*</code>	use text figures in text and math mode
<code>textlf</code>	use lining figures in text mode
<code>mathlf</code>	use lining figures in math mode
<code>lf</code>	use lining figures in text and math mode
<code>mathtabular</code>	use tabular figures in math mode

### Calligraphic fonts

These options specify which font is used by the `\mathcal` command.

<code>mnsy*</code>	use the calligraphic font from MnSymbol: $\mathcal{ABC}$
<code>cmsy</code>	take the calligraphic symbols from Computer Modern: $\mathcal{ABC}$
<code>swash</code>	use the swash capitals from MinionPro: $\mathcal{ABC}$
<code>abx</code>	use the calligraphic symbols provided by mathabx: $\mathcal{ABCabc}$ (This font contains also lowercase letters, but it is not quite finished.)

### Blackboard bold letters

You can also select different fonts for the `\mathbb` command.

<code>amsbb*</code>	use the AMS blackboard font: $\mathbb{NZQRC}$
<code>fourierbb</code>	use the Fourier blackboard font: $\mathbb{NZQRC}$
<code>lucidabb</code>	use the (commercial) Lucida Math blackboard font

## Greek letters

The following options specify whether you want to use upright or italic Greek letters in math mode.

<code>mixedgreek*</code>	uppercase Greek is upright, lowercase Greek is italic
<code>italicgreek</code>	all Greek letters are italic
<code>frenchmath</code>	all Greek letters and the uppercase Roman letters are upright

Upright and italic Greek letters are also directly accessible via the commands `\upgamma`, `\itgamma`, `\upGamma`, `\itGamma`, etc.

## Miscellaneous options

<code>scale=&lt;factor&gt;</code>	scale the font size by <i>&lt;factor&gt;</i>
<code>minionint</code>	take the integral symbols from MinionPro, not from MnSymbol: $\int$ instead of $\int$
<code>openg</code>	use <i>g</i> instead of <i>g</i> in math mode.
<code>loosequotes</code>	The quote signs of MinionPro are set rather tight. This can lead to undesirable spacing for apostrophes. The <code>loosequotes</code> option slightly increases the side bearings of quotes. This option requires pdfTeX 1.40 and microtype 2.0. Beware that this option prevents hyphenation of words containing apostrophes. Such words will require explicit hyphenation commands <code>\-</code> .
<code>footnotefigures</code>	use special figures for footnote marks, i.e., <code>example<sup>6,9</sup></code> instead of <code>example<sup>6,9</sup></code> . This option can only be used if the footnote marks consist <i>solely</i> of figures.

## 4 Figure selection

MinionPro offers four different figure versions. One can choose between *text figures* (lowercase figures) and *lining figures* (uppercase figures) and one can choose between *proportional* figures (figures with different widths) and *tabular* figures (all figures have the same width, useful mainly for tables).

	text figures	lining figures
proportional	o123456789	0123456789
tabular	o123456789	0123456789

The `\figureversion` command can be used to switch between different figure versions. Possible parameters are:

<code>text, osf</code>	text figures
<code>lining, lf</code>	lining figures

tabular, tab	tabular figures
proportional, prop	proportional figures

Usually it is desirable to set most text with proportional figures and to use tabular figures only in tables and lists. Unfortunately most  $\LaTeX$  document classes do not support fonts with several figure versions. Use the package `tabfigures` that patches some common document classes and packages (the standard  $\LaTeX$  classes, KOMA-Script, memoir, and amsmath) to use tabular figures at some places.

## 5 Additional font shapes and symbols

In addition to the normal small caps shape `sc` there is a letterspaced version called `ssc`. It is accessible via the commands `\sscshape` and `\textssc`. In order to use the `ssc` shape throughout your document specify `\renewcommand{\scdefault}{ssc}` in the preamble of your document.

Swash capitals like ‘*Canadian Mountain Holidays*’ are accessed via the `sw` fontshape and the commands `\swshape` and `\textsw`.

<code>sc</code>	THIS IS A SAMPLE TEXT
<code>ssc</code>	THIS IS A SAMPLE TEXT
<code>sw</code>	<i>This is a Sample Text</i>

The MinionPro package provides all symbols from the MnSymbol package. Additionally, the following math symbols are available:

$\digamma$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beta$	<code>\varbeta</code>
$\backepsilon$	<code>\backepsilon</code>	$\varbackepsilon$	<code>\varbackepsilon</code>	$\hbar$	<code>\hbar</code>
$\hslash$	<code>\hslash</code>	$\lambda\text{dabar}$	<code>\lambdabar</code>	$\lambda\text{slash}$	<code>\lambdaslash</code>
$\jmath$	<code>\jmath</code>	$\eth$	<code>\eth</code>	$\Bbbk$	<code>\Bbbk</code>
$\text{\slashed{0}}$	<code>\slashedzero</code>	$\text{\openg}$	<code>\openg</code>		

Small and slanted fractions are fractions with a height matching the font’s body size. These are useful for typesetting, e.g.,  $\cos(\frac{1}{2}x + \frac{3}{2}y)$  or “ $\frac{1}{12}$  litres of red wine” and can be accessed via

<code>\smallfrac{⟨numerator⟩}{⟨denominator⟩}</code>	$\frac{1}{3} \frac{5}{17}$
<code>\slantfrac{⟨numerator⟩}{⟨denominator⟩}</code>	$\frac{1}{3} \frac{5}{17}$

Note that *only* figures can be used for `⟨numerator⟩` and `⟨denominator⟩`.

Ornaments can be accessed via the `pifont` package with the command

`\Pisymbol{MinionPro-Extra}{⟨number⟩}`

The available glyphs are listed in the table below. Version 1.000 of the MinionPro font provides only ornaments 100–122.

number	glyph	number	glyph	number	glyph	number	glyph
100	Ⲁ	113	ⲛ	126	■	139	■
101	ⲁ	114	Ⲝ	127	■	140	■
102	Ⲃ	115	ⲝ	128	■	141	■
103	ⲃ	116	Ⲟ	129	■	142	■
104	Ⲅ	117	ⲟ	130	■	143	■
105	ⲅ	118	Ⲡ	131	■	144	■
106	Ⲇ	119	ⲡ	132	■	145	■
107	ⲇ	120	Ⲣ	133	■	146	■
108	Ⲉ	121	ⲣ	134	■	147	■
109	ⲉ	122	Ⲥ	135	■	148	■
110	Ⲋ	123	■	136	■	149	■
111	ⲋ	124	■	137	■	150	■
112	Ⲍ	125	■	138	■		

## 6 Language support

The following encodings are supported:

Latin	OT1, T1, TS1, LY1, T5
Cyrillic	T2A, T2B, T2C, X2, OT2
Greek	LGR (to be used with babel, including polutonikogreek), LGI (lbycus transliteration scheme)

In order to typeset Greek text with the lbycus transliteration scheme, specify

```
\usepackage[ibycus,<otherlanguages>]{babel}
```

in the preamble and consult the documentation given in `ibycus-babel.pdf` on CTAN.  
`\setgreekfontsize` is not supported.

## 7 Searching for figures or for words containing ligatures in PDF documents

Searching for figures or for words containing ligatures in PDF documents may not be possible depending on the way the PDF file was created. The following table gives an overview of which glyphs may cause problems.

font version	program	problems
1.000	Ghostscript, pre-1.40 pdf $\TeX$	LF/TOf, non-standard ligatures, swashes
1.001, 2.000	Ghostscript, pre-1.40 pdf $\TeX$	LF/OsF/TOf, ligatures, swashes, small caps
1.00x	Distiller, dvipdfmx	LF/TOf
1.00x	pdf $\TeX$ 1.40	ok
2.000	Distiller, dvipdfmx, pdf $\TeX$ 1.40	ok

To make figures and ligatures searchable when using pdf $\TeX$  1.40, you need to enable glyph-to-unicode translation and load the default mapping table:

```
\input glyptounicode
\pdfgentounicode=1
```

See the pdf $\TeX$  manual for details.

## 8 NFSS classification

Parenthesised combinations are provided via substitutions.

encoding	family	series	shape
OT1, T1, TS1, LY1, T5	MinionPro-OsF, MinionPro-LF, MinionPro-TOf, MinionPro-TLF	m, b (sb, bx), eb	n, it (sl), sw <sup>1</sup> , sc, scit (scsl, scsw), ssc, sscit (sscs, sscsw)
LGR, LGI, T2A, T2B, T2C, X2, OT2	MinionPro-OsF, MinionPro-LF, MinionPro-TOf, MinionPro-TLF	m, b (sb, bx), eb	n, it (sl)
OML	MinionPro-TOf	m, b (sb, bx), eb	n, it
U	MinionPro-Extra	m, b (sb, bx), eb	n, it (sl)

## 9 Version history

Version 2.0: Initial Release on CTAN

Version 2.1:

---

<sup>1</sup>via substitution in TS1 encoding

- added package options `onlytext` and `onlymath`
- added package option `loosequotes`
- added package option `openg`
- added package options `normalsize` and `nonnormalsize`
- fixed package option `frenchmath`
- fixed package option `abx`
- added support for pdfTeX 1.4 CMAP inclusion
- update to microtype version 1.8
- added `tabfigures` to automatically handle tabular figures in toc, equation labels, bibliographies, enumerations
- fixed `\t` accent
- fixed `\r` accent in OT1 encoding
- fixed slashed zero in font version 2.000
- fixed arrows in TS1 and U encodings
- fixed LGR and LGI encodings to use  $\phi$  instead of  $\phi$
- fixed 'P' in LGI encoding
- added punctuation support in LGI encoding (thanks to Jens Boerstinghaus)
- added symbols `\hslash`, `\lambdabar`, `\lambdaslash`
- fixed side bearings of  $\sigma$  in math mode
- added CODINGSCHEME statements to encoding files
- fixed usage of MnSymbol's "|" in doc.sty's module prefix
- reduce number of raw encodings to five per font

Version 2.2:

- add `scale` option
- fix typo in microtype hook

Version 2.3:

- fix<sup>2</sup> `footnotefigures` option with KOMA classes

## 10 The main style file

### 10.1 Options

```

1 \langle *style\rangle
2 \newif\if@Mn@Text@
3 \newif\if@Mn@Math@
4 \@Mn@Text@true
5 \@Mn@Math@true
6 \RequirePackage{kvoptions}
7 \SetupKeyvalOptions{
8   family = Mn,
9   prefix = Mn@
10 }
11 \DeclareVoidOption{onlytext}{\@Mn@Text@true\@Mn@Math@false}
12 \DeclareVoidOption{onlymath}{\@Mn@Text@false\@Mn@Math@true}

```

---

<sup>2</sup>based on <http://tex.stackexchange.com/a/54954/11605>



## Font sets

The package MinionPro-FontDef adapts the font definitions to the requested font set (see section 12). So we simply pass on the relevant options including the font scale factor; only MinionPro integrals are handled here in MinionPro.

```
13 \DeclareStringOption[1.]{scale}
14 \newcommand\Mn@minionint@opticals{-NoOpticals}
15 \newcommand\Mn@minionint@bold{-Bold}
16 \DeclareVoidOption{slides}{%
17   \def\Mn@minionint@opticals{-NoOpticals}%
18   \PassOptionsToPackage{slides}{MinionPro-FontDef}}
19 \DeclareVoidOption{noopticals}{%
20   \def\Mn@minionint@opticals{-NoOpticals}%
21   \PassOptionsToPackage{noopticals}{MinionPro-FontDef}}
22 \DeclareVoidOption{opticals}{%
23   \def\Mn@minionint@opticals{}%
24   \PassOptionsToPackage{opticals}{MinionPro-FontDef}}
25 \DeclareVoidOption{smallfamily}{%
26   \def\Mn@minionint@bold{-Bold}%
27   \PassOptionsToPackage{smallfamily}{MinionPro-FontDef}}
28 \DeclareVoidOption{medfamily}{%
29   \def\Mn@minionint@bold{-Semibold}%
30   \PassOptionsToPackage{medfamily}{MinionPro-FontDef}}
31 \DeclareVoidOption{fullfamily}{%
32   \def\Mn@minionint@bold{-Semibold}%
33   \PassOptionsToPackage{fullfamily}{MinionPro-FontDef}}
34 \DeclareVoidOption{normalsize}{%
35   \PassOptionsToPackage{normalsize}{MinionPro-FontDef}}
36 \DeclareVoidOption{nonnormalsize}{%
37   \PassOptionsToPackage{nonnormalsize}{MinionPro-FontDef}}
```

## Figure style

```
38 \newcommand\Mn@Text@Fig{OsF}
39 \newcommand\Mn@Math@Fig{OsF}
40 \newcommand\Mn@Text@Family{MinionPro-\Mn@Text@Fig}
41 \newcommand\Mn@Math@Family{MinionPro-\Mn@Math@Fig}
42 \newcommand\Mn@Math@TFamily{MinionPro-T\Mn@Math@Fig}
43 \newcommand\Mn@Math@LetterShape{it}
44 \DeclareVoidOption{textosf}{\def\Mn@Text@Fig{OsF}}
45 \DeclareVoidOption{textlf}{\def\Mn@Text@Fig{LF}}
46 \DeclareVoidOption{mathosf}{\def\Mn@Math@Fig{OsF}}
47 \DeclareVoidOption{mathlf}{\def\Mn@Math@Fig{LF}}
48 \DeclareVoidOption{osf}{\setkeys{Mn}{textosf,mathosf}}
49 \DeclareVoidOption{lf}{\setkeys{Mn}{textlf,mathlf}}
50 \DeclareVoidOption{mathtabular}{\let\Mn@Math@Family\Mn@Math@TFamily}
```

## Calligraphic fonts

These hooks are executed once the math versions have been set up.

```

51 \newcommand\Mn@load@cal{}
52 \newcommand\Mn@load@bb{}
53 \newcommand\Mn@load@frak{}

```

Most options are handled by MnSymbol.

```

54 \DeclareVoidOption{mnsy}{
55   \PassOptionsToPackage{mnsy}{MnSymbol}
56   \def\Mn@load@cal{
57     \SetMathAlphabet\mathcal{boldtabular}{OMS}{MnSymbolS}{b}{n}
58   }
59 }
60 \DeclareVoidOption{cmsy}{
61   \PassOptionsToPackage{cmsy}{MnSymbol}
62   \def\Mn@load@cal{
63     \SetMathAlphabet\mathcal{boldtabular}{OMS}{cmsy}{b}{n}
64   }
65 }
66 \DeclareVoidOption{abx}{
67   \PassOptionsToPackage{abx}{MnSymbol}
68 %   \def\Mn@load@cal{
69 %     \SetMathAlphabet\mathcal{boldtabular}{OT1}{mathc}{b}{n}
70 %   }
71 }
72 \DeclareVoidOption{swash}{
73   \def\Mn@load@cal{
74     \DeclareMathAlphabet\mathcal{T1}{\Mn@Math@Family}{m}{sw}
75     \SetMathAlphabet\mathcal{bold}{T1}{\Mn@Math@Family}{eb}{sw}
76     \SetMathAlphabet\mathcal{tabular}{T1}{\Mn@Math@TFamily}{m}{sw}
77     \SetMathAlphabet\mathcal{boldtabular}{T1}{\Mn@Math@TFamily}{eb}{sw}}
78 }

```

## Greek letters

`\Mn@greek@Upright`, `\Mn@greek@Mixed`, and `\Mn@greek@Italic` are defined below in section 10.4 before `\Mn@load@greek` is executed.

```

79 \newcommand\Mn@load@greek{\Mn@greek@Mixed}
80 \DeclareVoidOption{frenchmath}{%
81   \def\Mn@load@greek{\Mn@greek@Upright}%
82   \def\Mn@Math@LetterShape{n}}
83 \DeclareVoidOption{mixedgreek}{%
84   \def\Mn@load@greek{\Mn@greek@Mixed}}
85 \DeclareVoidOption{italicgreek}{%
86   \def\Mn@load@greek{\Mn@greek@Italic}}

```

## Blackboard bold and fraktur fonts

We have to undefine `\mathfrak` and `\mathbb` before redefining them, because they might be defined in such a way that `\DeclareMathAlphabet` does not recognize them

as math alphabets and refuses to overwrite their definitions (e.g., package eufrak uses `\newcommand{\mathfrak}{\EuFrak}`).

```

87 \newcommand\Mn@load@amsbb{
88   \let\mathbb\@undefined
89   \let\Bbbk\@undefined
90   \DeclareMathAlphabet\mathbb{U}{msb}{m}{n}
91   \newcommand\Bbbk{\mathbb{\mathchar"717C}}
92 \newcommand\Mn@load@lucidabb{
93   \let\mathbb\@undefined
94   \let\Bbbk\@undefined
95   \DeclareFontFamily{U}{hlcm}{}
96   \DeclareFontShape{U}{hlcm}{m}{n}{<->s*[0.92] hlcra }{}
97   \DeclareMathAlphabet\mathbb{U}{hlcm}{m}{n}
98   \newcommand\Bbbk{\mathbb{k}}
99 \newcommand\Mn@load@fourierbb{
100   \let\mathbb\@undefined
101   \let\Bbbk\@undefined
102   \DeclareFontFamily{U}{futm}{}
103   \DeclareFontShape{U}{futm}{m}{n}{<->s*[0.95] fourier-bb }{}
104   \DeclareMathAlphabet\mathbb{U}{futm}{m}{n}
105   \newcommand\Bbbk{\mathbb{k}}
106 \DeclareVoidOption{amsbb}{\let\Mn@load@bb\Mn@load@amsbb}
107 \DeclareVoidOption{lucidabb}{\let\Mn@load@bb\Mn@load@lucidabb}
108 \DeclareVoidOption{fourierbb}{\let\Mn@load@bb\Mn@load@fourierbb}

```

## Integrals

```

109 \newcommand\Mn@load@integrals{
110 \DeclareVoidOption{minionint}{\def\Mn@load@integrals{\Mn@Decl@Minion@Ints}}

```

## Miscellaneous options

Footnote figures, the g glyph in math mode, extra spacing for the apostrophe.

```

111 \DeclareVoidOption{footnotefigures}{%
112   \def\@makefnmark{%
113     \begingroup
114     \normalfont
115     \fontfamily{MinionPro-Extra}\fontencoding{U}\selectfont
116     \@thefnmark
117   \endgroup}%
118   \@ifundefined{KOMAClassName}{\def\footnote[1em]{1.5em}{1em}}{%
119     \fontfamily{MinionPro-Extra}\fontencoding{U}\selectfont\thefootnotemark}}
120 %
121 \newcommand\Mn@Define@Open@g{
122 \DeclareVoidOption{openg}{%
123   \def\Mn@Define@Open@g{%
124     \mathcode'g="8000%
125     \DeclareMathSymbol{\Mn@g}{\mathalpha}{letters}{'g}%
126     \begingroup
127     \lccode'\-='g

```

```

128 \lowercase{\gdef~{\ifnum\the\mathgroup=\m@ne \openg \else \Mn@g \fi}}%
129 \endgroup
130 }}
131 %
132 \newcommand\Mn@Quote@Spacing{}
133 \DeclareVoidOption{loosequotes}{%
134 \def\Mn@Quote@Spacing{\Mn@Quote@Spacing@Loose}}

```

## Defaults

```

135 \setkeys{Mn}{amsbb}
136 \ProcessKeyvalOptions{Mn}\relax

```

## 10.2 Font declarations

```

137 \RequirePackage{MinionPro-FontDef}
138 \@ifpackageloaded{textcomp}{\RequirePackage{textcomp}}
139
140 \if@Mn@Math@
141 \RequirePackage{MnSymbol}[2007/01/21 v1.4]

```

If no fraktur font is loaded then take the Euler font.

```

142 \@ifundefined{mathfrak}{%
143 \RequirePackage{eufrak}%
144 \SetMathAlphabet\EuFrak{boldtabular}{U}{euf}{b}{n}}{f}
145 \fi

```

By default, we use b for the bold series. If MinionPro-Semibold is not available this might internally be mapped to MinionPro-Bold (see MinionPro-FontDef).

```

146 \if@Mn@Text@
147 \edef\rmdefault{\Mn@Text@Family}
148 \let\ibycusdefault\Mn@Text@Family

```

If a recent version of microtype is loaded then we implement an option to increase the side bearings of all quote glyphs.

```

149 \def\Mn@Quote@Spacing@Loose{%
150 \@ifpackageloaded{microtype}{\RequirePackage[kerning=true]{microtype}}
151 \@ifundefined{SetExtraKerning}{%
152 \let\Mn@Set@Quote@Spacing\SetExtraKerning}
153 % \SetExtraKerning
154 % [ unit = 1em ]
155 % { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2,LY1},
156 % family = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
157 % shape = n }
158 % { \textquotedblleft = {30,30}, \textquotedblright = {30,30},
159 % \textquoteleft = {30,30}, \textquoteright = {30,30} }}
160 }
161 \newcommand*\Mn@Set@Quote@Spacing[3][]{%
162 \Mn@Quote@Spacing
163 \Mn@Set@Quote@Spacing
164 [ unit = 1em ]
165 { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2,LY1},

```

```

166     family    = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
167     shape      = {n,it} }
168 { \textquotedblleft = {30,30}, \textquotedblright = {30,30},
169   \textquoteleft   = {30,30}, \textquoteright    = {30,30} }
170 \fi

```

## Math fonts

Redefine the standard math versions normal and bold.

```

171 \if@Mn@Math@
172 \DeclareSymbolFont{operators} {T1} {\Mn@Math@Family}{m}{n}
173 \DeclareSymbolFont{letters}   {OML}{MinionPro-T0sF}{m}{\Mn@Math@LetterShape}
174 \SetSymbolFont{operators}{bold}{T1} {\Mn@Math@Family}{eb}{n}
175 \SetSymbolFont{letters}   {bold}{OML}{MinionPro-T0sF}{eb}{\Mn@Math@LetterShape}
176 \DeclareMathAlphabet\mathbf {T1} {\Mn@Math@Family}{eb}{n}
177 \DeclareMathAlphabet\mathit {T1} {\Mn@Math@Family}{m}{it}
178 \SetMathAlphabet\mathit {bold}{T1} {\Mn@Math@Family}{eb}{it}

```

Extra math versions tabular and boldtabular, which use tabular figures instead of proportional ones. These math versions can be useful in tables (cf. section 2).

```

179 \DeclareMathVersion{tabular}
180 \SetSymbolFont{operators}{tabular} {T1} {\Mn@Math@TFamily}{m}{n}
181 \SetSymbolFont{letters} {tabular}   {OML}{MinionPro-T0sF}{m}{\Mn@Math@LetterShape}
182 \SetMathAlphabet\mathit {tabular}   {T1} {\Mn@Math@TFamily}{m}{it}
183
184 \DeclareMathVersion{boldtabular}
185 \SetSymbolFont{operators}{boldtabular}{T1} {\Mn@Math@TFamily}{eb}{n}
186 \SetSymbolFont{letters} {boldtabular}{OML}{MinionPro-T0sF}{eb}{\Mn@Math@LetterShape}
187 \SetMathAlphabet\mathit {boldtabular}{T1} {\Mn@Math@TFamily}{eb}{it}
188
189 \DeclareMathAccent{\grave} {\mathalpha}{operators}{0}
190 \DeclareMathAccent{\acute} {\mathalpha}{operators}{1}
191 \DeclareMathAccent{\hat} {\mathalpha}{operators}{2}
192 \DeclareMathAccent{\tilde} {\mathalpha}{operators}{3}
193 \DeclareMathAccent{\ddot} {\mathalpha}{operators}{4}
194 \DeclareMathAccent{\mathring} {\mathalpha}{operators}{6}
195 \DeclareMathAccent{\check} {\mathalpha}{operators}{7}
196 \DeclareMathAccent{\breve} {\mathalpha}{operators}{8}
197 \DeclareMathAccent{\bar} {\mathalpha}{operators}{9}
198 \DeclareMathAccent{\dot} {\mathalpha}{operators}{10}

```

Execute the hooks set up above to load the various math alphabets.

```

198 \Mn@load@bb
199 \Mn@load@frak
200 \Mn@load@cal
201 \fi

```

## 10.3 Font selection

The font selection commands such as `\figureversion`, `\textsw`, and `\textssc` are provided by the package `fontaxes`.

```
202 \RequirePackage{fontaxes}[2005/05/04]
```

We define an additional short hand for compatibility's sake.

```
203 \let\oldstylenums\textfigures
```

## 10.4 Greek letters

We provide math-mode commands for each Greek letter, both italic and upright. Furthermore, there are three commands to select the default version of the letters (all upright, all italic, or capitals upright and lowercase italic).

While declaring the Greek letters we collect the uppercase and lowercase letters in two lists. (We distinguish them by the first letter of their name.) These lists are then used to select the different versions.

```
204 \if@Mn@Math@
205 \newcommand\Mn@greek@list@upper{}
206 \newcommand\Mn@greek@list@lower{}
207 \let\Mn@greek@list@upper\@gobble
208 \let\Mn@greek@list@lower\@gobble
```

This macro holds one of the two list names.

```
209 \newcommand\Mn@greek@list{}
210 \newcommand*\Mn@greek@letter[3]{%
211 \expandafter\DeclareMathSymbol
212 \expandafter{\csname it#1\endcsname}{\mathord}{letters}{#2}%
213 \expandafter\DeclareMathSymbol
214 \expandafter{\csname up#1\endcsname}{\mathord}{letters}{#3}%
215 \edef\@tempa{'\@car#1\@nil}%
216 \edef\Mn@greek@list{\expandafter\noexpand\csname
217 Mn@greek@list@\ifnum\uccode\@tempa=\@tempa upper\else lower\fi\endcsname}%
218 \expandafter\edef\Mn@greek@list{\Mn@greek@list,#1}%
219 }
```

We can now declare the Greek letters (left italic, right upright).

```
220 \Mn@greek@letter{Gamma}      {'000}{'200}
221 \Mn@greek@letter{Delta}      {'001}{'201}
222 \Mn@greek@letter{Theta}      {'002}{'202}
223 \Mn@greek@letter{Lambda}     {'003}{'203}
224 \Mn@greek@letter{Xi}         {'004}{'204}
225 \Mn@greek@letter{Pi}         {'005}{'205}
226 \Mn@greek@letter{Sigma}      {'006}{'206}
227 \Mn@greek@letter{Upsilon}    {'007}{'207}
228 \Mn@greek@letter{Phi}        {'010}{'210}
229 \Mn@greek@letter{Psi}        {'011}{'211}
230 \Mn@greek@letter{Omega}      {'012}{'212}
231 \Mn@greek@letter{alpha}      {'013}{'213}
232 \Mn@greek@letter{beta}       {'014}{'214}
233 \Mn@greek@letter{gamma}      {'015}{'215}
234 \Mn@greek@letter{delta}      {'016}{'216}
235 \Mn@greek@letter{epsilon}    {'017}{'217}
236 \Mn@greek@letter{zeta}       {'020}{'220}
```

```

237 \Mn@greek@letter{eta}          {'021}{'221}
238 \Mn@greek@letter{theta}        {'022}{'222}
239 \Mn@greek@letter{iota}          {'023}{'223}
240 \Mn@greek@letter{kappa}         {'024}{'224}
241 \Mn@greek@letter{lambda}        {'025}{'225}
242 \Mn@greek@letter{mu}            {'026}{'226}
243 \Mn@greek@letter{nu}            {'027}{'227}
244 \Mn@greek@letter{xi}            {'030}{'230}
245 \Mn@greek@letter{pi}            {'031}{'231}
246 \Mn@greek@letter{rho}           {'032}{'232}
247 \Mn@greek@letter{sigma}         {'033}{'233}
248 \Mn@greek@letter{tau}           {'034}{'234}
249 \Mn@greek@letter{upsilon}       {'035}{'235}
250 \Mn@greek@letter{phi}           {'036}{'236}
251 \Mn@greek@letter{chi}           {'037}{'237}
252 \Mn@greek@letter{psi}           {'040}{'240}
253 \Mn@greek@letter{omega}         {'041}{'241}
254 \Mn@greek@letter{varepsilon}    {'042}{'242}
255 \Mn@greek@letter{vartheta}      {'043}{'243}
256 \Mn@greek@letter{varpi}         {'044}{'244}
257 \Mn@greek@letter{varrho}        {'045}{'245}
258 \Mn@greek@letter{varsigma}     {'046}{'246}
259 \Mn@greek@letter{varphi}        {'047}{'247}

```

Some of the following symbols are not really Greek letters but are treated in the same way.

```

260 \Mn@greek@letter{varbeta}      {'260}{'250}
261 \Mn@greek@letter{varkappa}      {'261}{'251}
262 \Mn@greek@letter{backepsilon}   {'262}{'252}
263 \Mn@greek@letter{varbackepsilon}{'263}{'253}
264 \Mn@greek@letter{digamma}       {'264}{'254}
265 \Mn@greek@letter{eth}           {'266}{'256}

```

Go through a list #2 of Greek letters and \let them be their #1-prefixed variants.

```

266 \newcommand*\Mn@greek@select[2]{%
267   \expandafter\let\expandafter\Mn@greek@list\csname Mn@greek@list@#2\endcsname
268   \@for\@tempa:=\Mn@greek@list\do{%
269     \expandafter\let\csname\@tempa\expandafter\endcsname
270     \csname#1\@tempa\endcsname
271   }%
272 }
273 \newcommand*\Mn@greek@Upright{%
274   \Mn@greek@select{up}{upper}%
275   \Mn@greek@select{up}{lower}%
276 }
277 \newcommand*\Mn@greek@Italic{%
278   \Mn@greek@select{it}{upper}%
279   \Mn@greek@select{it}{lower}%
280 }
281 \newcommand*\Mn@greek@Mixed{%
282   \Mn@greek@select{up}{upper}%
283   \Mn@greek@select{it}{lower}%

```

```
284 }
```

Finally initialise the Greek letters.

```
285 \Mn@load@greek
286 \fi
```

## 10.5 pdfTeX to-unicode support

Old versions of MinionPro have non-standard glyph names.

```
287 \@ifundefined{pdfglyphtounicode}{-}{
288   \pdfglyphtounicode{uniEFD5}{03DD}% uni03DD
289   \pdfglyphtounicode{uniEFED}{02D9}% dotaccent.cap
290   \pdfglyphtounicode{uniEFEE}{02D8}% breve.cap
291   \pdfglyphtounicode{uniEFF1}{02DB}% ogonek.cap
292   \pdfglyphtounicode{uniEFF2}{00B8}% cedilla.cap
293   \pdfglyphtounicode{uniEFF3}{02DA}% ring.cap
294   \pdfglyphtounicode{uniEFF5}{02DC}% tilde.cap
295   \pdfglyphtounicode{uniEFF7}{02C6}% circumflex.cap
296   \pdfglyphtounicode{uniF628}{2030}% perthousand.oldstyle
297   \pdfglyphtounicode{uniF62C}{0028}% parenleft.denominator
298   \pdfglyphtounicode{uniF62D}{0029}% parenright.denominator
299   \pdfglyphtounicode{uniF631}{0028}% parenleft.numerator
300   \pdfglyphtounicode{uniF632}{0029}% parenright.numerator
301   \pdfglyphtounicode{uniF638}{0030}% zero.slash
302   \pdfglyphtounicode{uniF639}{0030}% zero.fitted
303   \pdfglyphtounicode{uniF63A}{0032}% two.fitted
304   \pdfglyphtounicode{uniF63B}{0033}% three.fitted
305   \pdfglyphtounicode{uniF63C}{0034}% four.fitted
306   \pdfglyphtounicode{uniF63D}{0035}% five.fitted
307   \pdfglyphtounicode{uniF63E}{0036}% six.fitted
308   \pdfglyphtounicode{uniF63F}{0037}% seven.fitted
309   \pdfglyphtounicode{uniF640}{0038}% eight.fitted
310   \pdfglyphtounicode{uniF641}{0039}% nine.fitted
311   \pdfglyphtounicode{uniF642}{0025}% percent.oldstyle
312   \pdfglyphtounicode{uniF643}{0030}% zero.taboldstyle
313   \pdfglyphtounicode{uniF644}{0031}% one.taboldstyle
314   \pdfglyphtounicode{uniF645}{0032}% two.taboldstyle
315   \pdfglyphtounicode{uniF646}{0033}% three.taboldstyle
316   \pdfglyphtounicode{uniF647}{0034}% four.taboldstyle
317   \pdfglyphtounicode{uniF648}{0035}% five.taboldstyle
318   \pdfglyphtounicode{uniF649}{0036}% six.taboldstyle
319   \pdfglyphtounicode{uniF64A}{0037}% seven.taboldstyle
320   \pdfglyphtounicode{uniF64B}{0038}% eight.taboldstyle
321   \pdfglyphtounicode{uniF64C}{0039}% nine.taboldstyle
322   \pdfglyphtounicode{uniF64D}{20A1}% colonmonetary.taboldstyle
323   \pdfglyphtounicode{uniF64E}{20AC}% Euro.taboldstyle
324   \pdfglyphtounicode{uniF64F}{0192}% florin.taboldstyle
325   \pdfglyphtounicode{uniF650}{0023}% numbersign.taboldstyle
326   \pdfglyphtounicode{uniF651}{00A3}% sterling.taboldstyle
327   \pdfglyphtounicode{uniF652}{00A5}% yen.taboldstyle
```



328 \pdfglyphtounicode{uniF653}{0024}% dollar.taboldstyle  
 329 \pdfglyphtounicode{uniF654}{00A2}% cent.taboldstyle  
 330 \pdfglyphtounicode{uniF655}{0030}% zero.denominator  
 331 \pdfglyphtounicode{uniF656}{0031}% one.denominator  
 332 \pdfglyphtounicode{uniF657}{0032}% two.denominator  
 333 \pdfglyphtounicode{uniF658}{0033}% three.denominator  
 334 \pdfglyphtounicode{uniF659}{0034}% four.denominator  
 335 \pdfglyphtounicode{uniF65A}{0035}% five.denominator  
 336 \pdfglyphtounicode{uniF65B}{0036}% six.denominator  
 337 \pdfglyphtounicode{uniF65C}{0037}% seven.denominator  
 338 \pdfglyphtounicode{uniF65D}{0038}% eight.denominator  
 339 \pdfglyphtounicode{uniF65E}{0039}% nine.denominator  
 340 \pdfglyphtounicode{uniF65F}{002C}% comma.denominator  
 341 \pdfglyphtounicode{uniF660}{002E}% period.denominator  
 342 \pdfglyphtounicode{uniF661}{0030}% zero.numerator  
 343 \pdfglyphtounicode{uniF662}{0031}% one.numerator  
 344 \pdfglyphtounicode{uniF663}{0032}% two.numerator  
 345 \pdfglyphtounicode{uniF664}{0033}% three.numerator  
 346 \pdfglyphtounicode{uniF665}{0034}% four.numerator  
 347 \pdfglyphtounicode{uniF666}{0035}% five.numerator  
 348 \pdfglyphtounicode{uniF667}{0036}% six.numerator  
 349 \pdfglyphtounicode{uniF668}{0037}% seven.numerator  
 350 \pdfglyphtounicode{uniF669}{0038}% eight.numerator  
 351 \pdfglyphtounicode{uniF66A}{0039}% nine.numerator  
 352 \pdfglyphtounicode{uniF66B}{002C}% comma.numerator  
 353 \pdfglyphtounicode{uniF66C}{002E}% period.numerator  
 354 \pdfglyphtounicode{uniF66D}{0103}% abreve.sc  
 355 \pdfglyphtounicode{uniF66F}{0105}% aogonek.sc  
 356 \pdfglyphtounicode{uniF671}{0107}% cacute.sc  
 357 \pdfglyphtounicode{uniF672}{010D}% ccaron.sc  
 358 \pdfglyphtounicode{uniF675}{010F}% dcaron.sc  
 359 \pdfglyphtounicode{uniF676}{0111}% dcroat.sc  
 360 \pdfglyphtounicode{uniF678}{011B}% ecaron.sc  
 361 \pdfglyphtounicode{uniF67B}{014B}% eng.sc  
 362 \pdfglyphtounicode{uniF67C}{0119}% eogonek.sc  
 363 \pdfglyphtounicode{uniF67D}{011F}% gbreve.sc  
 364 \pdfglyphtounicode{uniF684}{0133}% ij.sc  
 365 \pdfglyphtounicode{uniF687}{0129}% itilde.sc  
 366 \pdfglyphtounicode{uniF68A}{013A}% lacute.sc  
 367 \pdfglyphtounicode{uniF68B}{013E}% lcaron.sc  
 368 \pdfglyphtounicode{uniF68E}{0144}% nacute.sc  
 369 \pdfglyphtounicode{uniF68F}{0148}% ncaron.sc  
 370 \pdfglyphtounicode{uniF692}{0151}% ohungarumlaut.sc  
 371 \pdfglyphtounicode{uniF695}{0155}% racute.sc  
 372 \pdfglyphtounicode{uniF696}{0159}% rcaron.sc  
 373 \pdfglyphtounicode{uniF698}{015B}% sacute.sc  
 374 \pdfglyphtounicode{uniF699}{015F}% scedilla.sc  
 375 \pdfglyphtounicode{uniF69D}{0165}% tcaron.sc  
 376 \pdfglyphtounicode{uniF69E}{0163}% tcommaaccent.sc  
 377 \pdfglyphtounicode{uniF6A0}{0171}% uhungarumlaut.sc

```

378 \pdfglyphtounicode{uniF6A3}{016F}% uring.sc
379 \pdfglyphtounicode{uniF6A4}{0169}% utilde.sc
380 \pdfglyphtounicode{uniF6AA}{1EF3}% ygrave.sc
381 \pdfglyphtounicode{uniF6AB}{017A}% zacute.sc
382 \pdfglyphtounicode{uniF6AC}{017C}% zdotaccent.sc
383 \pdfglyphtounicode{uniF6DC}{0031}% one.fitted
384 }

```

## 10.6 Superior and inferior figures

We define commands to convert numbers to numerator figures and denominator figures.

```

385 \def\@for@tok#1:=#2\do#3{%
386   \expandafter\def\expandafter\@fortmp\expandafter{#2}%
387   \ifx\@fortmp\@empty \else
388     \expandafter\@forloop@tok#2\@nil\@nil\@@#1{#3}%
389   \fi}
390 \def\@forloop@tok#1#2#3\@@#4#5{%
391   \def#4{#1}%
392   \ifx #4\@nnil \else
393     #5%
394     \def#4{#2}%
395     \ifx #4\@nnil \else
396       #5\@forloop@tok #3\@@#4{#5}%
397     \fi\fi}
398 \def\@iforloop@tok#1#2\@@#3#4{%
399   \def#3{#1}%
400   \ifx #3\@nnil
401     \expandafter\@fornoop
402   \else
403     #4\relax\expandafter\@iforloop@tok
404   \fi
405   #2\@@#3{#4}}
406 %
407 \newcommand*\Mn@extra@font{%
408   \fontencoding{U}\fontfamily{MinionPro-Extra}\selectfont}
409 \newcommand*\@numerator@fig[1]{\Mn@extra@font\@@numerator@fig{#1}}
410 \newcommand*\@denominator@fig[1]{\Mn@extra@font\@@denominator@fig{#1}}
411 \newcommand*\@superior@fig[1]{\Mn@extra@font\@@superior@fig{#1}}
412 \newcommand*\@inferior@fig[1]{\Mn@extra@font\@@inferior@fig{#1}}
413 \newcommand*\@@numerator@fig[1]{%
414   \@for@tok\@nf@fig:=#1\do{%
415     \ifcase\@nf@fig
416       \char'00%
417     \or\char'01%
418     \or\char'02%
419     \or\char'03%
420     \or\char'04%
421     \or\char'05%
422     \or\char'06%

```

```

423 \or\char'07%
424 \or\char'10%
425 \or\char'11%
426 \else
427 \latex@error{invalid argument to \string\@@numerator@fig}%
428 \fi
429 }}
430 \newcommand*\@@denominator@fig[1]{%
431 \@for@tok\@nf@fig:=#1\do{%
432 \ifcase\@nf@fig
433 \char'20%
434 \or\char'21%
435 \or\char'22%
436 \or\char'23%
437 \or\char'24%
438 \or\char'25%
439 \or\char'26%
440 \or\char'27%
441 \or\char'30%
442 \or\char'31%
443 \else
444 \latex@error{invalid argument to \string\@@denominator@fig}%
445 \fi
446 }}
447 \newcommand*\@@superior@fig[1]{%
448 \@for@tok\@nf@fig:=#1\do{%
449 \ifcase\@nf@fig
450 \char'60%
451 \or\char'61%
452 \or\char'62%
453 \or\char'63%
454 \or\char'64%
455 \or\char'65%
456 \or\char'66%
457 \or\char'67%
458 \or\char'70%
459 \or\char'71%
460 \else
461 \latex@error{invalid argument to \string\@@superior@fig}%
462 \fi
463 }}
464 \newcommand*\@@inferior@fig[1]{%
465 \@for@tok\@nf@fig:=#1\do{%
466 \ifcase\@nf@fig
467 \char'100%
468 \or\char'101%
469 \or\char'102%
470 \or\char'103%
471 \or\char'104%
472 \or\char'105%

```

```

473 \or\char'106%
474 \or\char'107%
475 \or\char'110%
476 \or\char'111%
477 \else
478 \latex@error{invalid argument to \string\@@inferior@fig}%
479 \fi
480 }

```

\ensure@text switches to text mode, if necessary.

```

481 \newcommand*\ensure@text[1]{%
482 \ifmmode
483 \Mn@Text@With@MathVersion{#1}%
484 \else
485 #1%
486 \fi}

```

\smallfrac and \slantfrac assemble numerical fractions.

```

487 \newcommand*\smallfrac[2]{%
488 \leavevmode
489 \setbox\@tempboxa
490 \vbox{%
491 \baselineskip\z@skip%
492 \lineskip.25ex%
493 \lineskiplimit-\maxdimen
494 \ialign{\hfil##\hfil\cr
495 \vbox to 2.13ex{\vss\hbox{\@numerator@fig{#1}}\vskip.68ex}\cr
496 \leavevmode\leaders\hrule height 1.1ex depth -1.01ex\hfill\cr
497 \vtop to 1ex{\vbox{\hbox{\@denominator@fig{#2}}\vss}\cr
498 \noalign{\vskip-1.47ex}}}%
499 \dp\@tempboxa=0.49ex%
500 \box\@tempboxa}
501 \newcommand*\slantfrac[2]{%
502 {\Mn@extra@font\@numerator@fig{#1}\kern-0.05em/\kern-0.06em\@denominator@fig{#2}}}
503 \DeclareRobustCommand*\smallfrac[2]{\ensure@text{\kern0.06em\smallfrac{#1}{#2}\kern0.09em}}
504 \DeclareRobustCommand*\slantfrac[2]{\ensure@text{\kern0.06em\slantfrac{#1}{#2}\kern0.09em}}

```

## 10.7 Additional symbols

Some symbols missing from MnSymbol can be taken from MinionPro.

```

505 \if\Mn@Math@
506 \let\hbar\undefined
507 \DeclareMathSymbol{\hbar}{\mathord}{letters}{'265}
508 \DeclareMathSymbol{\uphbar}{\mathord}{letters}{'255}
509 \DeclareMathSymbol{\partial}{\mathord}{letters}{'100}
510 \DeclareMathSymbol{\uppartial}{\mathord}{letters}{'300}
511 \DeclareMathSymbol{\ell}{\mathord}{letters}{'140}
512 \DeclareMathSymbol{\upell}{\mathord}{letters}{'340}
513 \DeclareMathSymbol{\slashedzero}{\mathord}{letters}{'257}
514 \DeclareMathSymbol{\upimath}{\mathord}{letters}{'373}

```

```

515 \DeclareMathSymbol{\upjmath}          {\mathord}{letters}{'374}
516 \DeclareMathSymbol{\varsmallint}      {\mathord}{letters}{'376}
517 \DeclareMathSymbol{\openg}             {\mathalpha}{letters}{'267}
518 \DeclareRobustCommand\lambdabar        {\middlebar\lambda}
519 \DeclareRobustCommand\lambdaslash      {\middleslash\lambda}
520 \fi

```

Archaic Greek letters not provided by MinionPro.

```

521 \if@Mn@Text@
522 %\def\Qoppa{\reflectbox{P}}
523 %\def\Sampi{\begingroup\fontfamily{cmr}\fontencoding{LGR}\selectfont\char23\endgroup}
524 \let\Stigma\sigma
525
526 % fix \r A
527 \DeclareTextCompositeCommand{\r}{OT1}{A}
528 {\leavevmode\setbox\z@\hbox{!}\dimen@=\ht\z@\advance\dimen@-1ex%
529 \oalign{\hss\raise.67\dimen@\hbox{\char23}\hss\crrc A}}
530
531 \DeclareEncodingSubset{TS1}{MinionPro-LF} {1}%
532 \DeclareEncodingSubset{TS1}{MinionPro-TLF} {1}%
533 \DeclareEncodingSubset{TS1}{MinionPro-OsF} {1}%
534 \DeclareEncodingSubset{TS1}{MinionPro-TOsF}{1}%
535 \AtBeginDocument{
536 \UndeclareTextCommand{\textvisiblespace}{T1}%
537 \UndeclareTextCommand{\textcompwordmark}{T1}%
538 \UndeclareTextCommand{\textsterling}{T1}%
539 \UndeclareTextCommand{\j}{T1}%
540 \UndeclareTextCommand{\j}{LY1}%
541 }
542 \fi

```

## 10.8 Integral symbols

We can also replace the integral signs from MnSymbol by those of MinionPro. The following definitions provide this as an option.

```

543 \if@Mn@Math@
544 \newcommand\Mn@Decl@Minion@Ints{%

```

Replace MnSymbolF by MnSymbolFI.

```

545 \DeclareFontFamily{U}{MnSymbolFI}{\}
546 \DeclareFontShape{U}{MnSymbolFI}{m}{it}{\}
547 <-6> MnSymbolFI\Mn@minionint@opticals5
548 <6-7> MnSymbolFI\Mn@minionint@opticals6
549 <7-8> MnSymbolFI\Mn@minionint@opticals7
550 <8-9> MnSymbolFI\Mn@minionint@opticals8
551 <9-10> MnSymbolFI\Mn@minionint@opticals9
552 <10-12> MnSymbolFI\Mn@minionint@opticals10
553 <12-> MnSymbolFI\Mn@minionint@opticals12
554 \}{}
555 \DeclareFontShape{U}{MnSymbolFI}{b}{it}{\}

```

```

556      <-6> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals5
557      <6-7> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals6
558      <7-8> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals7
559      <8-9> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals8
560      <9-10> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals9
561      <10-12> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals10
562      <12-> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals12
563    }{}

564    \DeclareSymbolFont{symbols} {U}{MnSymbolFI}{m}{it}
565    \SetSymbolFont{symbols}{bold}{U}{MnSymbolFI}{b}{it}

```

Make the original integral symbols available as \var....

```

566    \let\varint\tint
567    \let\variint\tiint
568    \let\variiint\tiiint
569    \let\variiiint\tiiiint
570    \let\varidotsint\tidotsint
571    \let\varlandupint\tlandupint
572    \let\varlanddownint\tlanddownint
573    \let\varstrokedint\tstrokedint
574    \let\varoint\toint
575    \let\varoiint\tioint
576    \let\varrcircclerightint\trcircclerightint
577    \let\varlcircclerightint\tlcircclerightint
578    \let\varrcircleleftint\trcircleleftint
579    \let\varlcircleleftint\tlcircleleftint
580    \let\varsumint\tsumint

```

Replace the symbols with the new integrals.

```

581    \DeclareMathSymbol\tint          \mathop{symbols}{112}
582    \DeclareMathSymbol\tiint         \mathop{symbols}{114}
583    \DeclareMathSymbol\tiiint        \mathop{symbols}{116}
584    \DeclareMathSymbol\tiiiint       \mathop{symbols}{118}
585    \DeclareMathSymbol\tidotsint     \mathop{symbols}{120}
586    \DeclareMathSymbol\tlandupint    \mathop{symbols}{122}
587    \DeclareMathSymbol\tlanddownint  \mathop{symbols}{124}
588    \DeclareMathSymbol\tstrokedint   \mathop{symbols}{126}
589    \DeclareMathSymbol\toint         \mathop{symbols}{128}
590    \DeclareMathSymbol\tioint        \mathop{symbols}{130}
591    \DeclareMathSymbol\trcircclerightint\mathop{symbols}{132}
592    \DeclareMathSymbol\tlcircclerightint\mathop{symbols}{134}
593    \DeclareMathSymbol\trcircleleftint \mathop{symbols}{136}
594    \DeclareMathSymbol\tlcircleleftint \mathop{symbols}{138}
595    \DeclareMathSymbol\tsumint       \mathop{symbols}{140}
596    \let\intop\tint
597    \let\ointop\toint
598  }

599  \Mn@load@integrals
600 \fi

```

## 10.9 Open G support

We can replace the closed  $g$  with the open variant  $g$ . The following definitions provide this as an option.

```
601 \if@Mn@Math@
602   \Mn@Define@Open@g
603 \fi
```

## 10.10 Logos

Correct logos.

```
604 \if@Mn@Text@
605   \def\TeX{T\kern-.1667em\lower.4ex\hbox{E}\kern-.125emX\@}
606   \DeclareRobustCommand{\LaTeX}{L\kern-.32em%
607     {\sbox\z@ T%
608       \vbox to\ht\z@{\hbox{\check@mathfonts
609         \fontsize\sf@size\z@
610         \math@fontsfalse\selectfont
611         A}%
612         \vss}}%
613     }%
614     \kern-.15em%
615     \TeX}
616 \fi
```

## 10.11 AMS

Fix a bug in amsmath.sty which does not support math fonts without a skew char.

```
617 \def\macc@set@skewchar#1{%
618   \begingroup
619   \ifnum\mathgroup=\m@ne \let\@tempa\@ne
620   \else
621     \ifnum\skewchar\textfont\mathgroup=\m@ne \let\@tempa\@ne
622     \else \let\@tempa\mathgroup
623   \fi
624   \fi
625   \count@=\skewchar\textfont\@tempa
626   \ifnum\count@=\m@ne
627     \endgroup
628     \def\macc@skewchar{}
629   \else
630     \advance\count@"7100
631     \edef\@tempa{\endgroup
632       \mathchardef\noexpand\macc@skewchar=\number\count@\relax}%
633     \@tempa
634   \fi
635   #1%
636 }
```

Make the changes take effect. This concludes the main style file.

```
637 \if@Mn@Text@
638 \normalfont
639 \fi
640 \end{style}
```

## 11 Support for character protrusion

The microtype configuration. All four MinionPro families use the same file (cf. section 12).

```
641 \*mtcfg
642 \SetProtrusion
643 [ name = MinionPro-OT1-Roman ]
644 { encoding = OT1,
645   family = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
646   shape = n }
647 {
648   A = {40,40},
649   F = { ,60},
650   J = {90, },
651   K = { ,50},
652   L = { ,60},
653   T = {50,50},
654   V = {40,40},
655   W = {30,30},
656   X = {50,50},
657   Y = {50,50},
658   k = { ,60},
659   r = { ,80},
660   t = { ,100},
661   v = {70,70},
662   w = {40,40},
663   x = {60,60},
664   y = {70,70},
665   ! = {70,180},
666   ( = {60,30}, ) = {30,60},
667   [ = {100,160}, ] = {160,100},
668   {,} = {440,700},
669   . = {660,700},
670   : = {400,480},
671   ; = {350,440},
672   - = {700,700},
673   \textendash = {390,480}, \textemdash = {220,270},
674   \textquotedblleft = {380,250}, \textquotedblright = {250,380},
675   \textquoteleft = {670,450}, \textquoteright = {450,670},
676 }
677 \SetProtrusion
678 [ name = MinionPro-T1-Roman,
679   load = MinionPro-OT1-Roman ]
```



```

680 { encoding = T1,
681     family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOf,MinionPro-TLF},
682     shape    = n }
683 {
684     023 = { ,40}, % fft ligature
685     032 = { ,50}, % ft ligature
686     191 = {30,30}, % Th ligature
687     127 = {620,700}, % hyphen
688     \AE = {40, }, % AE
689     \quotesinglbase = {670,670}, \quotedblbase = {370,370},
690     \guilsinglleft = {500,360}, \guilsinglright = {360,500},
691     \guillemotleft = {320,230}, \guillemotright = {230,320},
692 }
693 \SetProtrusion
694 [ name      = MinionPro-OT1-Italic]
695 { encoding = OT1,
696     family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOf,MinionPro-TLF},
697     shape    = {it,sl,sw} }
698 {
699     A = {120,50},
700     B = {90,-50},
701     C = {50,-60},
702     D = {70,-30},
703     E = {90,-50},
704     F = {100,-40},
705     G = {50,-60},
706     H = {70,-40},
707     I = {150,-90},
708     J = {250,-130},
709     K = {80,-50},
710     L = {90,60},
711     M = {60,-40},
712     N = {70,-40},
713     O = {70,-30},
714     P = {70,-110},
715     Q = {40,-40},
716     R = {80,-50},
717     S = {70,-70},
718     T = {130, },
719     U = {70,-40},
720     V = {120,30},
721     W = {90,20},
722     X = {50, },
723     Y = {160, },
724     Z = {50,-50},
725     d = {60,-60},
726     f = { , -190},
727     027 = { , -70}, % ff ligature
728     g = {-70,-70},

```

```

729     i = { , -110},
730     025 = { , -60}, % dotlessi
731     028 = { , -60}, % fi ligature
732     030 = { , -30}, % ffi ligature
733     j = {-90, -150},
734     p = {-40, },
735     r = { , 80},
736     t = { , 100},
737     v = {90, },
738     w = {60, 10},
739     x = {90, },
740     ! = {190, 40},
741     ( = {90, }, ) = {90, },
742     [ = {90, 90}, ] = {120, 60},
743     {, } = {210, 680},
744     . = {640, 680},
745     : = {380, 430},
746     ; = { , 430},
747     - = {750, 750},
748     \textquoteleft = {690, 140}, \textquoteright = {470, 230},
749     \textendash = {400, 500}, \textemdash = {220, 280},
750     \textquotedblleft = {520, 130}, \textquotedblright = {520, 130},
751 }

752 \SetProtrusion
753 [ name = MinionPro-T1-Italic,
754   load = MinionPro-OT1-Italic ]
755 { encoding = T1,
756   family = {MinionPro-OsF, MinionPro-LF, MinionPro-T0sF, MinionPro-TLF},
757   shape = {it, sl, sw} }
758 {
759     023 = { , 40}, % fft ligature
760     032 = { , 50}, % ft ligature
761     191 = {80, 30}, % Th ligature
762     127 = {660, 750}, % hyphen
763     \AE = {90, -40}, % AE
764     131 = {80, -30}, % Dcaron
765     132 = {70, -40}, % Ecaron
766     156 = {80, -60}, % IJ
767     \OE = {50, -30}, % OE
768     188 = { , -80}, % ij
769     184 = {70, 70}, % ydieresis
770     253 = {70, 70}, % yacute
771     \quotesinglbase = {220, 700}, \quotedblbase = {130, 400},
772     \guilsinglleft = {500, 180}, \guilsinglright = {350, 350},
773     \guillemotleft = {310, 110}, \guillemotright = {230, 230},
774 }

```

We have no protruding values for small caps yet. The following stubs are unnecessary at the moment, but they are here as a reminder.

```

775 \SetProtrusion
776 [ name      = MinionPro-OT1-Smallcaps ]
777 { encoding = OT1,
778   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
779   shape     = {sc,ssc} }
780 {}

781 \SetProtrusion
782 [ name      = MinionPro-T1-Smallcaps,
783   load      = MinionPro-OT1-Smallcaps ]
784 { encoding = T1,
785   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
786   shape     = {sc,ssc} }
787 {}

788 \SetProtrusion
789 [ name      = MinionPro-OT1-SmallcapsItalic ]
790 { encoding = OT1,
791   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
792   shape     = {scit,sscit} }
793 {}

794 \SetProtrusion
795 [ name      = MinionPro-T1-SmallcapsItalic,
796   load      = MinionPro-OT1-SmallcapsItalic ]
797 { encoding = T1,
798   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
799   shape     = {scit,sscit} }
800 {}

801 \SetProtrusion
802 [ name      = MinionPro-other-Roman ]
803 { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
804   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
805   shape     = n }
806 {
807   ! = {70,180},
808   ( = {60,30},   ) = {30,60},
809   [ = {100,160}, ] = {160,100},
810   {,} = {440,700},
811   . = {660,700},
812   : = {400,480},
813   ; = {350,440},
814   - = {700,700},
815   \textendash      = {390,480},   \textemdash      = {220,270},
816   \textquotedblleft = {380,250},   \textquotedblright = {250,380},
817   \textquoteleft    = {670,450},   \textquoteright    = {450,670},
818 }

819 \SetProtrusion
820 [ name      = MinionPro-other-Italic ]
821 { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
822   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
823   shape     = {it,sl,sw} }

```

```

824 {
825     ! = {190,40},
826     ( = {90, }, ) = {90, },
827     [ = {90,90}, ] = {120,60},
828     {,} = {210,680},
829     . = {640,680},
830     : = {380,430},
831     ; = { ,430},
832     - = {750,750},
833     \textquoteleft = {690,140}, \textquoteright = {470,230},
834     \textendash = {400,500}, \textemdash = {220,280},
835     \textquotedblleft = {520,130}, \textquotedblright = {520,130},
836 }
837 \end{mtcfg}

```

## 12 Font definition files

As all the font definitions look the same we introduce macros to ease the configuration. These macros are stored in the file `MinionPro-FontDef.sty` which is included by every `FD` file. Note that `MinionPro-FontDef.sty` will be included several times and that we do not know in which context the code is executed. Therefore, we have to define all non-private commands as globals.

Since this package should be loadable in an `FD` file we have to avoid all `\preambleonly` commands. Therefore, we use `\ProvidesFile` instead of `\ProvidesPackage`.

We add a guard so that this file is executed only once even if it is included multiple times.

```

838 \fontdef
839 \ifx\Mn@DeclareFontShape\@undefined\else\endinput\fi

```

We distinguish between being loaded directly or via `\usepackage` in the preamble by checking `\@nodocument`.

```

840 \ifx\@nodocument\relax
841   \input{otfontdef.sty}
842 \else
843   \NeedsTeXFormat{LaTeX2e}
844   \RequirePackage{otfontdef}
845 \fi

```

Reset `\escapechar` (which is set to `-1` in `FD` files) to make `\newcommand` work. The additional group does not harm; we have to make the important commands global anyway.

```

846 \ifx\@nodocument\relax
847   \begingroup\escapechar'\
848 \fi

```

These are the default values if it is impossible to process options.

```

849 \newcommand\Mn@option@opticals{noopticals}
850 \newcommand\Mn@option@fontset{smallfamily}
851 \newdimen\Mn@option@normalsize
852 \global\Mn@option@normalsize10pt

```

Whether we should adapt the configuration to the `\normalsize` of the document. This switch is only needed locally.

```

853 \newif\ifMn@option@normalsize
854 \Mn@option@normalsizetrue

855 \ifx\@nodocument\relax\else
856   \DeclareOption{slides}      {\let\Mn@option@opticals\CurrentOption}
857   \DeclareOption{opticals}    {\let\Mn@option@opticals\CurrentOption}
858   \DeclareOption{noopticals} {\let\Mn@option@opticals\CurrentOption}
859   \DeclareOption{smallfamily} {\let\Mn@option@fontset\CurrentOption}
860   \DeclareOption{medfamily}   {\let\Mn@option@fontset\CurrentOption}
861   \DeclareOption{fullfamily}  {\let\Mn@option@fontset\CurrentOption}
862   \DeclareOption{normalsize}  {\Mn@option@normalsizetrue}
863   \DeclareOption{nonnormalsize} {\Mn@option@normalsizefalse}
864   \ExecuteOptions{smallfamily,noopticals,normalsize}
865   \ProcessOptions\relax
866 \fi

```

The method to determine the main font size is inspired by microtype’s implementation.

```

867 \ifMn@option@normalsize
868   \begingroup
869   \def\set@fontsize#1#2#3#4\@nil{%
870     \@defaultunits\global\Mn@option@normalsize#2pt\relax\@nnil}%
871   \normalsize\@nil
872   \endgroup
873 \fi

```

We use `\otf@makeglobal` from `otfontdef` to “export” the definitions that are needed globally.

```

874 \otf@makeglobal\Mn@option@opticals}
875 \otf@makeglobal\Mn@option@fontset}
876 \ifx\@nodocument\relax\else
877   \PackageInfo{MinionPro-FontDef}{%
878     Configuration:\space\Mn@option@fontset,\space\Mn@option@opticals,\space
879     normalsize=\the\Mn@option@normalsize}%
880 \fi

```

### Configuration database

```

881 \newcount\Mn@config@cnt
882 \Mn@config@cnt=0
883 \newcommand\Mn@curr@config{\Mn@config@romannumeral\Mn@config@cnt}

```

These commands help in setting up the configuration database. They do not need to be global. But the config database itself has to be.

#3 is added to all instances listed in #2 of configuration class #1. #3 is read with `NFSS` catcodes.

```

884 \newcommand\Mn@AddToConfig{%
885   \begingroup
886   \nfss@catcodes
887   \expandafter\endgroup

```

```

888 \Mn@AddToConfig@
889 }
890 \newcommand\Mn@AddToConfig@[3]{%
891 \advance\Mn@config@cnt\@ne
892 \@namedef{\Mn@curr@config}{#3}%
893 \otf@makeglobal{\Mn@curr@config}
894 <debug & show>\expandafter\show\csname\Mn@curr@config\endcsname
895 \@for\Mn@tempa:=#2\do{%
896 \@ifundefined{Mn@config@#1@\Mn@tempa}{%
897 \@temptokena{%
898 }{%
899 \@temptokena\expandafter\expandafter\expandafter
900 {\csname Mn@config@#1@\Mn@tempa\endcsname}%
901 }%
902 \@expandtwoargs\@namedef{Mn@config@#1@\Mn@tempa}{%
903 \the\@temptokena
904 \expandafter\noexpand\csname\Mn@curr@config\endcsname
905 }%
906 \otf@makeglobal{Mn@config@#1@\Mn@tempa}% perhaps defer to only execute once
907 <debug & show>\expandafter\show\csname Mn@config@#1@\Mn@tempa\endcsname
908 }%
909 }

```

Let us look at an example of how the configuration database looks internally for (shape, sw), which is specified below in three steps. The following lines show different depths of expansion of the macro \Mn@config@shape@sw, which finally yields the complete configuration:

```

\Mn@config@shape@sw
\Mn@config@xi \Mn@config@xiv \Mn@config@xv
<-8>otf*[spacing=11]<->otf*[variant=swash]<->otf*MinionPro-It

```

The following commands are used in the Declare...Family commands to access the previously built configuration database. They must be expandable. #3 is used as a default if no entry is found in the database.

```

910 \newcommand*\Mn@UseConfig[2]{%
911 \Mn@UseConfigOrDefault{#1}{#2}{}%
912 }
913 \newcommand*\Mn@UseConfigOrDefault[3]{%
914 \@ifundefined{Mn@config@#1@#2}{#3}%
915 {\@nameuse{Mn@config@#1@#2}}%
916 }
917 \newcommand*\Mn@TheConfig[2]{%
918 \@ifundefined{Mn@config@#1@#2}{}%
919 \expandafter\noexpand\csname Mn@config@#1@#2\endcsname
920 }%
921 }
922 \otf@makeglobal{Mn@UseConfig}
923 \otf@makeglobal{Mn@UseConfigOrDefault}
924 \otf@makeglobal{Mn@TheConfig}

```

The size range in the configuration has to be divided by the scaling factor to take the changed size into account because the scaling takes place after choosing the right combination. Provide calculation routine here.

```
925 \RequirePackage{fltpoint}
926 \fpDecimalSign{.}
927 \newcommand*{\Mn@calc@bsize}[2]{\fpDiv{#1}{#2}{\Mn@scale}}
```

Here comes the configuration.

```
928 \Mn@calc@bsize{\Mn@s@capt}{8.5}
929 \Mn@calc@bsize{\Mn@s@text}{13.1}
930 \Mn@calc@bsize{\Mn@s@subh}{20}
931 \Mn@AddToConfig{opticals}{opticals}{
932     <-\Mn@s@capt> otf* [optical=Capt]
933     <\Mn@s@capt-\Mn@s@text> otf* [optical=Text]
934     <\Mn@s@text-\Mn@s@subh> otf* [optical=Subh]
935     <\Mn@s@subh-> otf* [optical=Disp]
936 }
937 \Mn@AddToConfig{opticals}{noopticals}{
938     <-> otf* [optical=Text]
939 }
940 \Mn@AddToConfig{opticals}{slides}{
941     <-> otf* [optical=Capt]
942 }

943 \ifdim\Mn@option@normalsize<10.1pt
944     \Mn@calc@bsize{\Mn@s@semif}{6}
945     \Mn@calc@bsize{\Mn@s@medif}{8.5}
946 \else
947     \Mn@calc@bsize{\Mn@s@semif}{6}
948     \Mn@calc@bsize{\Mn@s@medif}{10.1}
949 \fi
950 \Mn@AddToConfig{fontset/weight}{fullfamily/m}{
951     <-\Mn@s@semif> otf* [weight=Semibold]
952     <\Mn@s@semif-\Mn@s@medif> otf* [weight=Medium]
953     <\Mn@s@medif-> otf* [weight=Regular]
954 }
955 \Mn@calc@bsize{\Mn@s@semim}{6}
956 \Mn@AddToConfig{fontset/weight}{medfamily/m}{
957     <-\Mn@s@semim> otf* [weight=Semibold]
958     <\Mn@s@semim-> otf* [weight=Regular]
959 }
960 \Mn@AddToConfig{fontset/weight}{smallfamily/m}{
961     <-> otf* [weight=Regular]
962 }
963 %
964 \Mn@calc@bsize{\Mn@s@bold}{6}
965 \Mn@AddToConfig{fontset/weight}{fullfamily/b,medfamily/b}{
966     <-\Mn@s@bold> otf* [weight=Bold]
967     <\Mn@s@bold-> otf* [weight=Semibold]
968 }
```

```

969 \Mn@AddToConfig{fontset/weight}{smallfamily/b}{
970     <->      otf* [weight=Bold]
971 }
972 %
973 \Mn@AddToConfig{weight}{eb}{
974     <->      otf* [weight=Bold]
975 }
976 \Mn@AddToConfig{shape}{ssc,sscit}{
977     <->      otf* [spacing=12]
978 }
979 \Mn@calc@bsize{\Mn@s@spac}{8}
980 \Mn@AddToConfig{shape}{n,it,sw,sc,scit}{
981     <-\Mn@s@spac>      otf* [spacing=11]
982 }
983 \Mn@AddToConfig{encoding/shape}{U/n,U/it}{
984     <->      otf* [spacing=]
985 }
986 %
987 \Mn@AddToConfig{shape}{sc,ssc,scit,sscit}{
988     <->      otf* [variant=sc]
989 }
990 \Mn@AddToConfig{shape}{sw}{
991     <->      otf* [variant=swash]
992 }
993 \Mn@AddToConfig{shape}{it,scit,sscit,sw}{
994     <->      otf* MinionPro-It
995 }
996 \Mn@AddToConfig{shape}{n,sc,ssc}{
997     <->      otf* MinionPro
998 }
999 \Mn@AddToConfig{encoding/shape}{OML/it}{
1000     <->      otf* [figures=] MinionPro-Mixed
1001 }
1002 \Mn@AddToConfig{encoding/shape}{OML/n}{
1003     <->      otf* [figures=] MinionPro-French
1004 }
1005 \Mn@AddToConfig{scale}{scale}{
1006     <->      otf* [scale=\Mn@scale]
1007 }

```

#### Substitutions

```

1008 \Mn@AddToConfig{sub:series} {sb}      {b}
1009 \Mn@AddToConfig{sub:series} {bx}      {b}
1010 \Mn@AddToConfig{sub:shape}   {sl}      {it}
1011 \Mn@AddToConfig{sub:shape}   {scsl}    {scit}
1012 \Mn@AddToConfig{sub:shape}   {sscs}    {sscit}
1013 \Mn@AddToConfig{sub:shape}   {scsw}    {scit}
1014 \Mn@AddToConfig{sub:shape}   {sscs}    {sscit}
1015 \Mn@AddToConfig{sub:encoding/shape}{TS1/sw}{it}

```



Code for the last argument of \DeclareFontShape

```
1016 \Mn@AddToConfig{code:shape}{sw}{
1017   \skewchar\font='337
1018 }
```

### Declaration of font families and shapes

```
1019 \newcommand*\Mn@DeclareFontShape[6] [] {%
```

Check if any substitutions are specified.

```
1020   \edef\@tempa{%
1021     \Mn@UseConfig{sub:series}{#4}%
1022     \Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
1023       \Mn@UseConfig{sub:shape}{#5}}}%
1024   }%
1025   \ifx\@tempa\@empty
```

Collect the configuration and declare the font shape. \DeclareFontShape fully expands its fifth argument (with our macros \Mn@UseConfig in it), but we have to retrieve the code for the sixth argument ourselves.

```
1026   \@temptokena={%
1027     \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
1028       \Mn@UseConfig{opticals}      {\Mn@option@opticals}%
1029       \Mn@UseConfig{fontset/weight}{\Mn@option@fontset/#4}%
1030       \Mn@UseConfig{weight}        {#4}%
1031       \Mn@UseConfig{encoding/shape}{#2/#5}%
1032       \Mn@UseConfig{shape}          {#5}%
1033       \Mn@UseConfig{scale}          {scale}%
1034     }}%
1035   \edef\@tempa{\the\@temptokena{\Mn@TheConfig{code:shape}{#5}}}%
1036   \@tempa
1037   \else
```

Generate the substitution. (All substitutions are silent at the moment.)

```
1038   \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
1039     <->ssub*#3-#6%
1040     /\Mn@UseConfigOrDefault{sub:series}{#4}{#4}%
1041     /\Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
1042       \Mn@UseConfigOrDefault{sub:shape}{#5}{#5}}%
1043   }{}%
1044   \fi
1045 }
```

```
1046 \otf@makeglobal\Mn@DeclareFontShape}
```

```
1047 \otf@makeglobal{\string\Mn@DeclareFontShape}
```

#2 contains the encoding, #3 the family, and #1 a list of figure versions (or Extra).

```
1048 \newcommand*\Mn@DeclareLargeFontFamily[3] [LF,OsF,TLF,TOfF]{%
1049   \Mn@DeclareFontFamily{#1}{#2}{#3}
1050   {m,sb,b,bx,eb} {n,it,sc,ssc,scit,sscit,sw,scsl,scsw,sscs,sl}%
1051 }
1052 \newcommand*\Mn@DeclareSmallFontFamily[3] [LF,OsF,TLF,TOfF]{%
1053   \Mn@DeclareFontFamily{#1}{#2}{#3}
```

```

1054 {m,sb,b,bx,eb} {n,it,sl}%
1055 }
1056 \newcommand*\Mn@DeclareMathFontFamily[3][T0sF]{%
1057 \Mn@DeclareFontFamily[\skewchar\font=255]{#1}{#2}{#3}
1058 {m,sb,b,bx,eb} {n,it}%
1059 }

```

An additional macro `\csname\string\foo\endcsname` is generated by `\newcommand` for processing an optional argument of `\foo`.

```

1060 \otf@makeglobal{\Mn@DeclareLargeFontFamily}
1061 \otf@makeglobal{\string\Mn@DeclareLargeFontFamily}
1062 \otf@makeglobal{\Mn@DeclareSmallFontFamily}
1063 \otf@makeglobal{\string\Mn@DeclareSmallFontFamily}
1064 \otf@makeglobal{\Mn@DeclareMathFontFamily}
1065 \otf@makeglobal{\string\Mn@DeclareMathFontFamily}
1066 \newcommand*\Mn@DeclareFontFamily[6][ ]{%
1067 \@for\Mn@variant:=#2\do{%
1068 \DeclareFontFamily {#3}{#4-\Mn@variant}{#1}%
1069 }%
1070 \Mn@DeclareFontShapes{#3}{#4}
1071 {#5} {#6} {#2}%
1072 }
1073 \otf@makeglobal{\Mn@DeclareFontFamily}
1074 \otf@makeglobal{\string\Mn@DeclareFontFamily}
1075 \newcommand*\Mn@DeclareFontShapes[5]{%
1076 \@for\Mn@series:=#3\do{%
1077 \@for\Mn@shape:=#4\do{%
1078 \@for\Mn@variant:=#5\do{%
1079 \Mn@DeclareFontShape{#1}{#2}{\Mn@series}{\Mn@shape}{\Mn@variant}%
1080 }%
1081 }%
1082 }%
1083 }
1084 \otf@makeglobal{\Mn@DeclareFontShapes}

```

Adjust font dimension #1 of the current font. The function in #2 should replace the old value in `\Mn@fontdimen` with a new one (which may depend on other parameters like `\f@size`).

```

1085 \newdimen\Mn@fontdimen
1086 \newcommand*\Mn@adjust@fontdimen[2]{%
1087 \Mn@fontdimen=\fontdimen#1\font
1088 #2%
1089 \fontdimen#1\font=\Mn@fontdimen
1090 }
1091 \otf@makeglobal{\Mn@adjust@fontdimen}
1092 \ifx\@nodocument\relax
1093 \endgroup
1094 \fi
1095 <*debug>

```

```

1096 \newcommand\old@DeclareFontFamily{}
1097 \let\old@DeclareFontFamily\DeclareFontFamily
1098 \renewcommand\DeclareFontFamily[3]{
1099   \begingroup\escapechar'\%
1100   \edef\@tempa{\noexpand\DeclareFontFamily{#1}{#2}}%
1101   \@temptokena\expandafter{\@tempa{#3}}%
1102   \message{\the\@temptokena}%
1103   \endgroup
1104   \old@DeclareFontFamily{#1}{#2}{#3}%
1105 }
1106 \newcommand\old@DeclareFontShape{}
1107 \let\old@DeclareFontShape\DeclareFontShape
1108 \renewcommand\DeclareFontShape[6]{
1109   \begingroup\escapechar'\%
1110   \edef\@tempa{\noexpand\DeclareFontShape{#1}{#2}{#3}{#4}{#5}}%
1111   \@temptokena\expandafter{\@tempa{#6}}%
1112   \message{\the\@temptokena}%
1113   \endgroup
1114   \old@DeclareFontShape{#1}{#2}{#3}{#4}{#5}{#6}%
1115 }
1116 </debug>

```

We define font family aliases so that we can place all configurations for the MinionPro family variants into one microtype file: `mt-MinionPro.cfg`. We use microtype's hook if microtype has not been loaded yet (which should be the case); otherwise we can execute the alias definitions directly.

```

1117 \gdef\Mn@MicroType@Aliases{%
1118   \DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}%
1119   \DeclareMicrotypeAlias{MinionPro-OsF}{MinionPro}%
1120   \DeclareMicrotypeAlias{MinionPro-TLF}{MinionPro}%
1121   \DeclareMicrotypeAlias{MinionPro-TOfS}{MinionPro}%
1122 }
1123 \@ifundefined{Microtype@Hook}{%
1124   \global\let\Microtype@Hook\Mn@MicroType@Aliases
1125 }{%
1126   \g@addto@macro\Microtype@Hook{\Mn@MicroType@Aliases}%
1127 }%
1128 \@ifundefined{DeclareMicroTypeAlias}{\Mn@MicroType@Aliases}%
1129 </fontdef>

```

Using these macros the various FD files become simple one-liners.

```

1130 < *fd>
1131 \input{MinionPro-FontDef.sty}%
1132 <Uextra> \Mn@DeclareSmallFontFamily[Extra]{U} {MinionPro}
1133 <LGR> \Mn@DeclareSmallFontFamily {LGR}{MinionPro}
1134 <LGI> \Mn@DeclareSmallFontFamily {LGI}{MinionPro}
1135 <OT1> \Mn@DeclareLargeFontFamily {OT1}{MinionPro}
1136 <T1> \Mn@DeclareLargeFontFamily {T1} {MinionPro}
1137 <LY1> \Mn@DeclareLargeFontFamily {LY1}{MinionPro}
1138 <T5> \Mn@DeclareLargeFontFamily {T5} {MinionPro}
1139 <T2A> \Mn@DeclareSmallFontFamily {T2A}{MinionPro}

```

```

1140 <T2B>          \Mn@DeclareSmallFontFamily      {T2B}{MinionPro}
1141 <T2C>          \Mn@DeclareSmallFontFamily      {T2C}{MinionPro}
1142 <TS1>          \Mn@DeclareLargeFontFamily       {TS1}{MinionPro}
1143 <X2>           \Mn@DeclareSmallFontFamily       {X2} {MinionPro}
1144 <OT2>          \Mn@DeclareSmallFontFamily       {OT2}{MinionPro}
1145 <OML & tosf>    \Mn@DeclareMathFontFamily       {OML}{MinionPro}
1146 <*OML & (lf | osf | tlf)>
1147   \@for\Mn@variant:=LF,TLF,OsF\do{%
1148     \DeclareFontFamily{OML}{MinionPro-\Mn@variant}{\skewchar\font=255}
1149     \@for\Mn@series:=m,sb,b,bx,eb\do{%
1150       \@for\Mn@shape:=n,it\do{%
1151         \DeclareFontShape{OML}{MinionPro-\Mn@variant}{\Mn@series}{\Mn@shape}%
1152           { <-> ssub*MinionPro-TOf/\Mn@series/\Mn@shape }{}
1153       }%
1154     }%
1155   }%
1156 </OML & (lf | osf | tlf)>
1157 </fd>

```