

# 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	22
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>	$\lambda\mkern-1mu\bar{\mkern-1mu}$	<code>\lambdabar</code>	$\lambda\mkern-1mu\text{\textsl{slash}}$	<code>\lambdaslash</code>
$\jmath$	<code>\jmath</code>	$\eth$	<code>\eth</code>	$\Bbbk$	<code>\Bbbk</code>
$\text{\textsl{zero}}$	<code>\slashedzero</code>	$\text{\textsl{open g}}$	<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

## 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}

```

#### 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}{hlc}{m}{n}{ <->s*[0.92] hlcra }{}
96 \DeclareFontShape{U}{hlc}{m}{n}{ <->s*[0.92] fourier-bb }{}
97 \DeclareMathAlphabet\mathbb{U}{hlc}{m}{n}
98 \newcommand\Bbbk{\mathbb{k}}
99 \newcommand\Mn@load@fourierbb{
100 \let\mathbb\@undefined
101 \let\Bbbk\@undefined
102 \DeclareFontFamily{U}{fut}{m}{n}{ <->s*[0.95] fourier-bb }{}
103 \DeclareFontShape{U}{fut}{m}{n}{ <->s*[0.95] fourier-bb }{}
104 \DeclareMathAlphabet\mathbb{U}{fut}{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 %
119 \newcommand\Mn@Define@Open@g{}
120 \DeclareVoidOption{openg}{%
121 \def\Mn@Define@Open@g{%
122 \mathcode'g="8000%
123 \DeclareMathSymbol{\Mn@g}{\mathalpha}{letters}{'g}%
124 \begingroup
125 \lccode'\~='g
126 \lowercase{\gdef~{\ifnum\the\mathgroup=\m@ne \openg \else \Mn@g \fi}}%
127 \endgroup
128 }}
129 %
130 \newcommand\Mn@Quote@Spacing{}
131 \DeclareVoidOption{loosequotes}{%

```

```
132 \def\Mn@Quote@Spacing{\Mn@Quote@Spacing@Loose}}
```

## Defaults

```
133 \setkeys{Mn}{\amsbb}
134 \ProcessKeyvalOptions{Mn}\relax
```

## 10.2 Font declarations

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

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

```
140 \@ifundefined{mathfrak}{%
141 \RequirePackage{eufrak}%
142 \SetMathAlphabet\EuFrak{boldtabular}{U}{euf}{b}{n}}{
143 \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).

```
144 \if@Mn@Text@
145 \edef\rmdefault{\Mn@Text@Family}
146 \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.

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

## Math fonts

Redefine the standard math versions normal and bold.

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

```
177 \DeclareMathVersion{tabular}
178 \SetSymbolFont{operators}{tabular} {T1} {\Mn@Math@TFamily}{m}{n}
179 \SetSymbolFont{letters} {tabular} {OML}{MinionPro-T0sF} {m}{\Mn@Math@LetterShape}
180 \SetMathAlphabet\mathit {tabular} {T1} {\Mn@Math@TFamily}{m}{it}
181
182 \DeclareMathVersion{boldtabular}
183 \SetSymbolFont{operators}{boldtabular}{T1} {\Mn@Math@TFamily}{eb}{n}
184 \SetSymbolFont{letters} {boldtabular}{OML}{MinionPro-T0sF} {eb}{\Mn@Math@LetterShape}
185 \SetMathAlphabet\mathit {boldtabular}{T1} {\Mn@Math@TFamily}{eb}{it}

186 \DeclareMathAccent{\grave} {\mathalpha}{operators}{0}
187 \DeclareMathAccent{\acute} {\mathalpha}{operators}{1}
188 \DeclareMathAccent{\hat} {\mathalpha}{operators}{2}
189 \DeclareMathAccent{\tilde} {\mathalpha}{operators}{3}
190 \DeclareMathAccent{\ddot} {\mathalpha}{operators}{4}
191 \DeclareMathAccent{\mathring} {\mathalpha}{operators}{6}
192 \DeclareMathAccent{\check} {\mathalpha}{operators}{7}
193 \DeclareMathAccent{\breve} {\mathalpha}{operators}{8}
194 \DeclareMathAccent{\bar} {\mathalpha}{operators}{9}
195 \DeclareMathAccent{\dot} {\mathalpha}{operators}{10}
```

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

```
196 \Mn@load@bb
197 \Mn@load@frak
198 \Mn@load@cal
199 \fi
```

## 10.3 Font selection

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

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

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

```
201 \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.

```

202 \if@Mn@Math@
203   \newcommand\Mn@greek@list@upper{}
204   \newcommand\Mn@greek@list@lower{}
205   \let\Mn@greek@list@upper\@gobble
206   \let\Mn@greek@list@lower\@gobble

```

This macro holds one of the two list names.

```

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

```

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

```

218   \Mn@greek@letter{\Gamma}      {'000}{'200}
219   \Mn@greek@letter{\Delta}      {'001}{'201}
220   \Mn@greek@letter{\Theta}      {'002}{'202}
221   \Mn@greek@letter{\Lambda}     {'003}{'203}
222   \Mn@greek@letter{\Xi}         {'004}{'204}
223   \Mn@greek@letter{\Pi}         {'005}{'205}
224   \Mn@greek@letter{\Sigma}      {'006}{'206}
225   \Mn@greek@letter{\Upsilon}    {'007}{'207}
226   \Mn@greek@letter{\Phi}        {'010}{'210}
227   \Mn@greek@letter{\Psi}        {'011}{'211}
228   \Mn@greek@letter{\Omega}      {'012}{'212}
229   \Mn@greek@letter{\alpha}      {'013}{'213}
230   \Mn@greek@letter{\beta}       {'014}{'214}
231   \Mn@greek@letter{\gamma}      {'015}{'215}
232   \Mn@greek@letter{\delta}      {'016}{'216}
233   \Mn@greek@letter{\epsilon}    {'017}{'217}
234   \Mn@greek@letter{\zeta}       {'020}{'220}
235   \Mn@greek@letter{\eta}        {'021}{'221}
236   \Mn@greek@letter{\theta}      {'022}{'222}
237   \Mn@greek@letter{\iota}       {'023}{'223}
238   \Mn@greek@letter{\kappa}      {'024}{'224}
239   \Mn@greek@letter{\lambda}     {'025}{'225}

```

```

240 \Mn@greek@letter{mu}          {'026}{'226}
241 \Mn@greek@letter{nu}          {'027}{'227}
242 \Mn@greek@letter{xi}          {'030}{'230}
243 \Mn@greek@letter{pi}          {'031}{'231}
244 \Mn@greek@letter{rho}          {'032}{'232}
245 \Mn@greek@letter{sigma}        {'033}{'233}
246 \Mn@greek@letter{tau}          {'034}{'234}
247 \Mn@greek@letter{upsilon}      {'035}{'235}
248 \Mn@greek@letter{phi}          {'036}{'236}
249 \Mn@greek@letter{chi}          {'037}{'237}
250 \Mn@greek@letter{psi}          {'040}{'240}
251 \Mn@greek@letter{omega}        {'041}{'241}
252 \Mn@greek@letter{varepsilon}    {'042}{'242}
253 \Mn@greek@letter{vartheta}      {'043}{'243}
254 \Mn@greek@letter{varpi}         {'044}{'244}
255 \Mn@greek@letter{varrho}         {'045}{'245}
256 \Mn@greek@letter{varsigma}      {'046}{'246}
257 \Mn@greek@letter{varphi}        {'047}{'247}

```

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

```

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

```

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

```

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

```

Finally initialise the Greek letters.

```

283 \Mn@load@greek
284 \fi

```

## 10.5 pdf<sub>TEX</sub> to-unicode support

Old versions of MinionPro have non-standard glyph names.

```
285 \ifundefined{pdfglyphtounicode}{%
286   \pdfglyphtounicode{uniEFD5}{03DD}% uni03DD
287   \pdfglyphtounicode{uniEFED}{02D9}% dotaccent.cap
288   \pdfglyphtounicode{uniEFEE}{02D8}% breve.cap
289   \pdfglyphtounicode{uniEFF1}{02DB}% ogonek.cap
290   \pdfglyphtounicode{uniEFF2}{00B8}% cedilla.cap
291   \pdfglyphtounicode{uniEFF3}{02DA}% ring.cap
292   \pdfglyphtounicode{uniEFF5}{02DC}% tilde.cap
293   \pdfglyphtounicode{uniEFF7}{02C6}% circumflex.cap
294   \pdfglyphtounicode{uniF628}{2030}% perthousand.oldstyle
295   \pdfglyphtounicode{uniF62C}{0028}% parenleft.denominator
296   \pdfglyphtounicode{uniF62D}{0029}% parenright.denominator
297   \pdfglyphtounicode{uniF631}{0028}% parenleft.numerator
298   \pdfglyphtounicode{uniF632}{0029}% parenright.numerator
299   \pdfglyphtounicode{uniF638}{0030}% zero.slash
300   \pdfglyphtounicode{uniF639}{0030}% zero.fitted
301   \pdfglyphtounicode{uniF63A}{0032}% two.fitted
302   \pdfglyphtounicode{uniF63B}{0033}% three.fitted
303   \pdfglyphtounicode{uniF63C}{0034}% four.fitted
304   \pdfglyphtounicode{uniF63D}{0035}% five.fitted
305   \pdfglyphtounicode{uniF63E}{0036}% six.fitted
306   \pdfglyphtounicode{uniF63F}{0037}% seven.fitted
307   \pdfglyphtounicode{uniF640}{0038}% eight.fitted
308   \pdfglyphtounicode{uniF641}{0039}% nine.fitted
309   \pdfglyphtounicode{uniF642}{0025}% percent.oldstyle
310   \pdfglyphtounicode{uniF643}{0030}% zero.taboldstyle
311   \pdfglyphtounicode{uniF644}{0031}% one.taboldstyle
312   \pdfglyphtounicode{uniF645}{0032}% two.taboldstyle
313   \pdfglyphtounicode{uniF646}{0033}% three.taboldstyle
314   \pdfglyphtounicode{uniF647}{0034}% four.taboldstyle
315   \pdfglyphtounicode{uniF648}{0035}% five.taboldstyle
316   \pdfglyphtounicode{uniF649}{0036}% six.taboldstyle
317   \pdfglyphtounicode{uniF64A}{0037}% seven.taboldstyle
318   \pdfglyphtounicode{uniF64B}{0038}% eight.taboldstyle
319   \pdfglyphtounicode{uniF64C}{0039}% nine.taboldstyle
320   \pdfglyphtounicode{uniF64D}{20A1}% colonmonetary.taboldstyle
321   \pdfglyphtounicode{uniF64E}{20AC}% Euro.taboldstyle
322   \pdfglyphtounicode{uniF64F}{0192}% florin.taboldstyle
323   \pdfglyphtounicode{uniF650}{0023}% numbersign.taboldstyle
324   \pdfglyphtounicode{uniF651}{00A3}% sterling.taboldstyle
325   \pdfglyphtounicode{uniF652}{00A5}% yen.taboldstyle
326   \pdfglyphtounicode{uniF653}{0024}% dollar.taboldstyle
327   \pdfglyphtounicode{uniF654}{00A2}% cent.taboldstyle
328   \pdfglyphtounicode{uniF655}{0030}% zero.denominator
329   \pdfglyphtounicode{uniF656}{0031}% one.denominator
330   \pdfglyphtounicode{uniF657}{0032}% two.denominator
331   \pdfglyphtounicode{uniF658}{0033}% three.denominator
```



332 \pdfglyphtounicode{uniF659}{0034}% four.denominator  
 333 \pdfglyphtounicode{uniF65A}{0035}% five.denominator  
 334 \pdfglyphtounicode{uniF65B}{0036}% six.denominator  
 335 \pdfglyphtounicode{uniF65C}{0037}% seven.denominator  
 336 \pdfglyphtounicode{uniF65D}{0038}% eight.denominator  
 337 \pdfglyphtounicode{uniF65E}{0039}% nine.denominator  
 338 \pdfglyphtounicode{uniF65F}{002C}% comma.denominator  
 339 \pdfglyphtounicode{uniF660}{002E}% period.denominator  
 340 \pdfglyphtounicode{uniF661}{0030}% zero.numerator  
 341 \pdfglyphtounicode{uniF662}{0031}% one.numerator  
 342 \pdfglyphtounicode{uniF663}{0032}% two.numerator  
 343 \pdfglyphtounicode{uniF664}{0033}% three.numerator  
 344 \pdfglyphtounicode{uniF665}{0034}% four.numerator  
 345 \pdfglyphtounicode{uniF666}{0035}% five.numerator  
 346 \pdfglyphtounicode{uniF667}{0036}% six.numerator  
 347 \pdfglyphtounicode{uniF668}{0037}% seven.numerator  
 348 \pdfglyphtounicode{uniF669}{0038}% eight.numerator  
 349 \pdfglyphtounicode{uniF66A}{0039}% nine.numerator  
 350 \pdfglyphtounicode{uniF66B}{002C}% comma.numerator  
 351 \pdfglyphtounicode{uniF66C}{002E}% period.numerator  
 352 \pdfglyphtounicode{uniF66D}{0103}% abreve.sc  
 353 \pdfglyphtounicode{uniF66F}{0105}% aogonek.sc  
 354 \pdfglyphtounicode{uniF671}{0107}% cacute.sc  
 355 \pdfglyphtounicode{uniF672}{010D}% ccaron.sc  
 356 \pdfglyphtounicode{uniF675}{010F}% dcaron.sc  
 357 \pdfglyphtounicode{uniF676}{0111}% dcroat.sc  
 358 \pdfglyphtounicode{uniF678}{011B}% ecaron.sc  
 359 \pdfglyphtounicode{uniF67B}{014B}% eng.sc  
 360 \pdfglyphtounicode{uniF67C}{0119}% eogonek.sc  
 361 \pdfglyphtounicode{uniF67D}{011F}% gbreve.sc  
 362 \pdfglyphtounicode{uniF684}{0133}% ij.sc  
 363 \pdfglyphtounicode{uniF687}{0129}% itilde.sc  
 364 \pdfglyphtounicode{uniF68A}{013A}% lacute.sc  
 365 \pdfglyphtounicode{uniF68B}{013E}% lcaron.sc  
 366 \pdfglyphtounicode{uniF68E}{0144}% nacute.sc  
 367 \pdfglyphtounicode{uniF68F}{0148}% ncaron.sc  
 368 \pdfglyphtounicode{uniF692}{0151}% ohungarumlaut.sc  
 369 \pdfglyphtounicode{uniF695}{0155}% racute.sc  
 370 \pdfglyphtounicode{uniF696}{0159}% rcaron.sc  
 371 \pdfglyphtounicode{uniF698}{015B}% sacute.sc  
 372 \pdfglyphtounicode{uniF699}{015F}% scedilla.sc  
 373 \pdfglyphtounicode{uniF69D}{0165}% tcaron.sc  
 374 \pdfglyphtounicode{uniF69E}{0163}% tcommaaccent.sc  
 375 \pdfglyphtounicode{uniF6A0}{0171}% uhungarumlaut.sc  
 376 \pdfglyphtounicode{uniF6A3}{016F}% uring.sc  
 377 \pdfglyphtounicode{uniF6A4}{0169}% utilde.sc  
 378 \pdfglyphtounicode{uniF6AA}{1EF3}% ygrave.sc  
 379 \pdfglyphtounicode{uniF6AB}{017A}% zacute.sc  
 380 \pdfglyphtounicode{uniF6AC}{017C}% zdotaccent.sc  
 381 \pdfglyphtounicode{uniF6DC}{0031}% one.fitted

```
382 }
```

## 10.6 Superior and inferior figures

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

```
383 \def\@for@tok#1:=#2\do#3{%
384   \expandafter\def\expandafter\@fortmp\expandafter{#2}%
385   \ifx\@fortmp\empty \else
386     \expandafter\@forloop@tok#2\@nil\@nil\@@#1{#3}%
387   \fi}
388 \def\@forloop@tok#1#2#3\@@#4#5{%
389   \def#4{#1}%
390   \ifx #4\@nnil \else
391     #5%
392     \def#4{#2}%
393     \ifx #4\@nnil \else
394       #5\@iforloop@tok #3\@@#4{#5}%
395     \fi\fi}
396 \def\@iforloop@tok#1#2\@@#3#4{%
397   \def#3{#1}%
398   \ifx #3\@nnil
399     \expandafter\@fornoop
400   \else
401     #4\relax\expandafter\@iforloop@tok
402   \fi
403   #2\@@#3{#4}}
404 %
405 \newcommand*\Mn@extra@font{%
406   \fontencoding{U}\fontfamily{MinionPro-Extra}\selectfont}
407 \newcommand*\@numerator@fig[1]{\{\Mn@extra@font\@@numerator@fig{#1}\}}
408 \newcommand*\@denominator@fig[1]{\{\Mn@extra@font\@@denominator@fig{#1}\}}
409 \newcommand*\@superior@fig[1]{\{\Mn@extra@font\@@superior@fig{#1}\}}
410 \newcommand*\@inferior@fig[1]{\{\Mn@extra@font\@@inferior@fig{#1}\}}
411 \newcommand*\@@numerator@fig[1]{%
412   \@for@tok\@nf@fig:=#1\do{%
413     \ifcase\@nf@fig
414       \char'00%
415     \or\char'01%
416     \or\char'02%
417     \or\char'03%
418     \or\char'04%
419     \or\char'05%
420     \or\char'06%
421     \or\char'07%
422     \or\char'10%
423     \or\char'11%
424     \else
425       \@latex@error{invalid argument to \string\@@numerator@fig}%
426     \fi
```

```

427     }}
428 \newcommand*{\@@denominator@fig[1]{%
429   \@for@tok\@nf@fig:=#1\do{%
430     \ifcase\@nf@fig
431       \char'20%
432     \or\char'21%
433     \or\char'22%
434     \or\char'23%
435     \or\char'24%
436     \or\char'25%
437     \or\char'26%
438     \or\char'27%
439     \or\char'30%
440     \or\char'31%
441     \else
442       \@latex@error{invalid argument to \string\@@denominator@fig}%
443     \fi
444   }}
445 \newcommand*{\@@superior@fig[1]{%
446   \@for@tok\@nf@fig:=#1\do{%
447     \ifcase\@nf@fig
448       \char'60%
449     \or\char'61%
450     \or\char'62%
451     \or\char'63%
452     \or\char'64%
453     \or\char'65%
454     \or\char'66%
455     \or\char'67%
456     \or\char'70%
457     \or\char'71%
458     \else
459       \@latex@error{invalid argument to \string\@@superior@fig}%
460     \fi
461   }}
462 \newcommand*{\@@inferior@fig[1]{%
463   \@for@tok\@nf@fig:=#1\do{%
464     \ifcase\@nf@fig
465       \char'100%
466     \or\char'101%
467     \or\char'102%
468     \or\char'103%
469     \or\char'104%
470     \or\char'105%
471     \or\char'106%
472     \or\char'107%
473     \or\char'110%
474     \or\char'111%
475     \else
476       \@latex@error{invalid argument to \string\@@inferior@fig}%

```

```

477 \fi
478 }}
\ensure@text switches to text mode, if necessary.
479 \newcommand*\ensure@text[1]{%
480 \ifmmode
481 \Mn@Text@With@MathVersion{#1}%
482 \else
483 #1%
484 \fi}
\smallfrac and \slantfrac assemble numerical fractions.
485 \newcommand*\@smallfrac[2]{%
486 \leavevmode
487 \setbox\@tempboxa
488 \vbox{%
489 \baselineskip\z@skip%
490 \lineskip.25ex%
491 \lineskiplimit-\maxdimen
492 \ialign{\hfil##\hfil\crcr
493 \vbox to 2.13ex{\vss\hbox{\@numerator@fig{#1}}\vskip.68ex}\crcr
494 \leavevmode\leaders\hrule height 1.1ex depth -1.01ex\hfill\crcr
495 \vtop to 1ex{\vbox{\hbox{\@denominator@fig{#2}}\vss}\crcr
496 \noalign{\vskip-1.47ex}}}%
497 \dp\@tempboxa=0.49ex%
498 \box\@tempboxa}
499 \newcommand*\@slantfrac[2]{%
500 {\Mn@extra@font\@numerator@fig{#1}\kern-0.05em\kern-0.06em\@denominator@fig{#2}}
501 \DeclareRobustCommand*\smallfrac[2]{\ensure@text{\kern0.06em\@smallfrac{#1}{#2}\kern0.09em}}
502 \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.

```

503 \if\Mn@Math@
504 \let\hbar\undefined
505 \DeclareMathSymbol{\hbar}{\mathord}{letters}{'265}
506 \DeclareMathSymbol{\uphbar}{\mathord}{letters}{'255}
507 \DeclareMathSymbol{\partial}{\mathord}{letters}{'100}
508 \DeclareMathSymbol{\uppartial}{\mathord}{letters}{'300}
509 \DeclareMathSymbol{\ell}{\mathord}{letters}{'140}
510 \DeclareMathSymbol{\upell}{\mathord}{letters}{'340}
511 \DeclareMathSymbol{\slashedzero}{\mathord}{letters}{'257}
512 \DeclareMathSymbol{\upimath}{\mathord}{letters}{'373}
513 \DeclareMathSymbol{\upjmath}{\mathord}{letters}{'374}
514 \DeclareMathSymbol{\varsmallint}{\mathord}{letters}{'376}
515 \DeclareMathSymbol{\openg}{\mathalpha}{letters}{'267}
516 \DeclareRobustCommand\lambdabar{\middlebar\lambda}
517 \DeclareRobustCommand\lambdaslash{\middleslash\lambda}
518 \fi

```

Archaic Greek letters not provided by MinionPro.

```

519 \if@Mn@Text@
520 %\def\Qoppa{\reflectbox{P}}
521 %\def\Sampi{\begingroup\fontfamily{cmr}\fontencoding{LGR}\selectfont\char23\endgroup}
522 \let\Stigma\sigma
523
524 % fix \r A
525 \DeclareTextCompositeCommand{\r}{OT1}{A}
526 {\leavevmode\setbox\z@\hbox{!}\dimen@ \ht\z@\advance\dimen@-1ex%
527 \oalign{\hss\raise.67\dimen@\hbox{\char23}\hss\crrc A}}
528
529 \DeclareEncodingSubset{TS1}{MinionPro-LF} {1}%
530 \DeclareEncodingSubset{TS1}{MinionPro-TLF} {1}%
531 \DeclareEncodingSubset{TS1}{MinionPro-OsF} {1}%
532 \DeclareEncodingSubset{TS1}{MinionPro-TOsF}{1}%
533 \AtBeginDocument{
534 \UndeclareTextCommand{\textvisiblespace}{T1}%
535 \UndeclareTextCommand{\textcompwordmark}{T1}%
536 \UndeclareTextCommand{\textsterling}{T1}%
537 \UndeclareTextCommand{\j}{T1}%
538 \UndeclareTextCommand{\j}{LY1}%
539 }
540 \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.

```

541 \if@Mn@Math@
542 \newcommand\Mn@Decl@Minion@Ints{%

```

Replace MnSymbolF by MnSymbolFI.

```

543 \DeclareFontFamily{U}{MnSymbolFI}{}
544 \DeclareFontShape{U}{MnSymbolFI}{m}{it}{
545 <-6> MnSymbolFI\Mn@minionint@opticals5
546 <6-7> MnSymbolFI\Mn@minionint@opticals6
547 <7-8> MnSymbolFI\Mn@minionint@opticals7
548 <8-9> MnSymbolFI\Mn@minionint@opticals8
549 <9-10> MnSymbolFI\Mn@minionint@opticals9
550 <10-12> MnSymbolFI\Mn@minionint@opticals10
551 <12-> MnSymbolFI\Mn@minionint@opticals12
552 }{}
553 \DeclareFontShape{U}{MnSymbolFI}{b}{it}{
554 <-6> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals5
555 <6-7> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals6
556 <7-8> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals7
557 <8-9> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals8
558 <9-10> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals9
559 <10-12> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals10

```

```

560     <12-> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals12
561 }{}
562 \DeclareSymbolFont{symbols} {U}{MnSymbolFI}{m}{it}
563 \SetSymbolFont{symbols}{bold}{U}{MnSymbolFI}{b}{it}

```

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

```

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

```

Replace the symbols with the new integrals.

```

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

```

599 \if@Mn@Math@

```

```

600 \Mn@Define@Open@g
601 \fi

```

## 10.10 Logos

Correct logos.

```

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

```

## 10.11 AMS

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

```

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

```

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

```

635 \if@Mn@Text@
636 \normalfont
637 \fi
638 \end{style}

```

## 11 Support for character protrusion

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

```

639 (*mtcfg)
640 \SetProtrusion
641 [ name      = MinionPro-OT1-Roman ]
642 { encoding = OT1,
643   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
644   shape     = n }
645 {
646   A = {40,40},
647   F = { ,60},
648   J = {90, },
649   K = { ,50},
650   L = { ,60},
651   T = {50,50},
652   V = {40,40},
653   W = {30,30},
654   X = {50,50},
655   Y = {50,50},
656   k = { ,60},
657   r = { ,80},
658   t = { ,100},
659   v = {70,70},
660   w = {40,40},
661   x = {60,60},
662   y = {70,70},
663   ! = {70,180},
664   ( = {60,30},      ) = {30,60},
665   [ = {100,160},    ] = {160,100},
666   {,} = {440,700},
667   . = {660,700},
668   : = {400,480},
669   ; = {350,440},
670   - = {700,700},
671   \textendash      = {390,480},   \textemdash      = {220,270},
672   \textquotedblleft = {380,250}, \textquotedblright = {250,380},
673   \textquoteleft    = {670,450},  \textquoteright  = {450,670},
674 }
675 \SetProtrusion
676 [ name      = MinionPro-T1-Roman,
677   load      = MinionPro-OT1-Roman ]
678 { encoding = T1,
679   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
680   shape     = n }
681 {
682   023 = { ,40}, % fft ligature
683   032 = { ,50}, % ft ligature
684   191 = {30,30}, % Th ligature

```



```

685 127 = {620,700}, % hyphen
686 \AE = {40, }, % AE
687 \quotesinglbase = {670,670}, \quotedblbase = {370,370},
688 \guilsinglleft = {500,360}, \guilsinglright = {360,500},
689 \guillemotleft = {320,230}, \guillemotright = {230,320},
690 }

691 \SetProtrusion
692 [ name = MinionPro-OT1-Italic]
693 { encoding = OT1,
694   family = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
695   shape = {it,sl,sw} }
696 {
697   A = {120,50},
698   B = {90,-50},
699   C = {50,-60},
700   D = {70,-30},
701   E = {90,-50},
702   F = {100,-40},
703   G = {50,-60},
704   H = {70,-40},
705   I = {150,-90},
706   J = {250,-130},
707   K = {80,-50},
708   L = {90,60},
709   M = {60,-40},
710   N = {70,-40},
711   O = {70,-30},
712   P = {70,-110},
713   Q = {40,-40},
714   R = {80,-50},
715   S = {70,-70},
716   T = {130, },
717   U = {70,-40},
718   V = {120,30},
719   W = {90,20},
720   X = {50, },
721   Y = {160, },
722   Z = {50,-50},
723   d = {60,-60},
724   f = { , -190},
725   027 = { , -70}, % ff ligature
726   g = {-70,-70},
727   i = { , -110},
728   025 = { , -60}, % dotlessi
729   028 = { , -60}, % fi ligature
730   030 = { , -30}, % ffi ligature
731   j = {-90,-150},
732   p = {-40, },
733   r = { , 80},

```

```

734     t = { ,100},
735     v = {90, },
736     w = {60,10},
737     x = {90, },
738     ! = {190,40},
739     ( = {90, }, ) = {90, },
740     [ = {90,90}, ] = {120,60},
741     {,} = {210,680},
742     . = {640,680},
743     : = {380,430},
744     ; = { ,430},
745     - = {750,750},
746     \textquoteleft = {690,140}, \textquoteright = {470,230},
747     \textendash = {400,500}, \textemdash = {220,280},
748     \textquotedblleft = {520,130}, \textquotedblright = {520,130},
749 }

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

```

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

```

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

779 \SetProtrusion

```

```

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

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

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

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

817 \SetProtrusion
818 [ name      = MinionPro-other-Italic ]
819 { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
820   family    = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
821   shape     = {it,sl,sw} }
822 {
823   ! = {190,40},
824   ( = {90,  },    ) = {90,  },
825   [ = {90,90},    ] = {120,60},
826   {,} = {210,680},
827   . = {640,680},
828   : = {380,430},

```

```

829      ; = { ,430},
830      - = {750,750},
831      \textquoteleft = {690,140}, \textquoteright = {470,230},
832      \textendash = {400,500}, \textemdash = {220,280},
833      \textquotedblleft = {520,130}, \textquotedblright = {520,130},
834  }
835 \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.

```

836 \fontdef
837 \ifx\Mn@DeclareFontShape\@undefined\else\endinput\fi

```

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

```

838 \ifx\@nodocument\relax
839   \input{otfontdef.sty}
840 \else
841   \NeedsTeXFormat{LaTeX2e}
842   \RequirePackage{otfontdef}
843 \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.

```

844 \ifx\@nodocument\relax
845   \begingroup\escapechar'\
846 \fi

```

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

```

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

```

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

```

851 \newif\ifMn@option@normalsize
852 \Mn@option@normalsizetrue

853 \ifx\@nodocument\relax\else
854   \DeclareOption{slides}    {\let\Mn@option@opticals\CurrentOption}
855   \DeclareOption{opticals} {\let\Mn@option@opticals\CurrentOption}

```

```

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

```

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

```

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

```

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

```

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

```

### Configuration database

```

879 \newcount\Mn@config@cnt
880 \Mn@config@cnt=0
881 \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.

```

882 \newcommand\Mn@AddToConfig{%
883 \begingroup
884 \nfss@catcodes
885 \expandafter\endgroup
886 \Mn@AddToConfig@
887 }
888 \newcommand\Mn@AddToConfig@[3]{%
889 \advance\Mn@config@cnt\@ne
890 \@namedef{\Mn@curr@config}{#3}%
891 \otf@makeglobal{\Mn@curr@config}
892 <debug & show>\expandafter\show\csname\Mn@curr@config\endcsname
893 \@for\Mn@tempa:=#2\do{%

```

```

894 \@ifundefined{Mn@config@#1@Mn@tempa}{%
895 \temptokena{}%
896 }{%
897 \temptokena\expandafter\expandafter\expandafter
898 {\csname Mn@config@#1@Mn@tempa\endcsname}%
899 }%
900 \@expandtwoargs\@namedef{Mn@config@#1@Mn@tempa}{%
901 \the\temptokena
902 \expandafter\noexpand\csname Mn@curr@config\endcsname
903 }%
904 \otf@makeglobal{Mn@config@#1@Mn@tempa}% perhaps defer to only execute once
905 <debug & show>\expandafter\show\csname Mn@config@#1@Mn@tempa\endcsname
906 }%
907 }

```

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.

```

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

```

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

```

Here comes the configuration.

```

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

941 \ifdim\Mn@option@normalsize<10.1pt
942 \Mn@calc@bsize{\Mn@s@semif}{6}
943 \Mn@calc@bsize{\Mn@s@medif}{8.5}
944 \else
945 \Mn@calc@bsize{\Mn@s@semif}{6}
946 \Mn@calc@bsize{\Mn@s@medif}{10.1}
947 \fi
948 \Mn@AddToConfig{fontset/weight}{fullfamily/m}{
949     < -\Mn@s@semif> otf* [weight=Semibold]
950     <\Mn@s@semif-\Mn@s@medif> otf* [weight=Medium]
951     <\Mn@s@medif-> otf* [weight=Regular]
952 }
953 \Mn@calc@bsize{\Mn@s@semim}{6}
954 \Mn@AddToConfig{fontset/weight}{medfamily/m}{
955     <-\Mn@s@semim> otf* [weight=Semibold]
956     <\Mn@s@semim-> otf* [weight=Regular]
957 }
958 \Mn@AddToConfig{fontset/weight}{smallfamily/m}{
959     <-> otf* [weight=Regular]
960 }
961 %
962 \Mn@calc@bsize{\Mn@s@bold}{6}
963 \Mn@AddToConfig{fontset/weight}{fullfamily/b,medfamily/b}{
964     <-\Mn@s@bold> otf* [weight=Bold]
965     <\Mn@s@bold-> otf* [weight=Semibold]
966 }
967 \Mn@AddToConfig{fontset/weight}{smallfamily/b}{
968     <-> otf* [weight=Bold]
969 }
970 %
971 \Mn@AddToConfig{weight}{eb}{
972     <-> otf* [weight=Bold]
973 }

974 \Mn@AddToConfig{shape}{ssc,sscit}{

```

```

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

```

#### Substitutions

```

1006 \Mn@AddToConfig{sub:series} {sb}      {b}
1007 \Mn@AddToConfig{sub:series} {bx}      {b}
1008 \Mn@AddToConfig{sub:shape} {sl}      {it}
1009 \Mn@AddToConfig{sub:shape} {scsl}    {scit}
1010 \Mn@AddToConfig{sub:shape} {sscs1}   {sscit}
1011 \Mn@AddToConfig{sub:shape} {scsw}    {scit}
1012 \Mn@AddToConfig{sub:shape} {sscsw}   {sscit}
1013 \Mn@AddToConfig{sub:encoding/shape}{TS1/sw}{it}

```

Code for the last argument of \DeclareFontShape

```

1014 \Mn@AddToConfig{code:shape}{sw}{
1015     \skewchar\font='337
1016 }

```

#### Declaration of font families and shapes

```

1017 \newcommand*\Mn@DeclareFontShape[6] [] {%

```



Check if any substitutions are specified.

```

1018 \edef\@tempa{%
1019     \Mn@UseConfig{sub:series}{#4}%
1020     \Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
1021         \Mn@UseConfig{sub:shape}{#5}}}%
1022 }%
1023 \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.

```

1024 \@temptokena={%
1025     \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
1026         \Mn@UseConfig{opticals}{\Mn@option@opticals}%
1027         \Mn@UseConfig{fontset/weight}{\Mn@option@fontset/#4}%
1028         \Mn@UseConfig{weight}{#4}%
1029         \Mn@UseConfig{encoding/shape}{#2/#5}%
1030         \Mn@UseConfig{shape}{#5}%
1031         \Mn@UseConfig{scale}{scale}%
1032     }}%
1033 \edef\@tempa{\the\@temptokena{\Mn@TheConfig{code:shape}{#5}}}%
1034 \@tempa
1035 \else

```

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

```

1036 \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
1037     <->ssub*#3-#6%
1038     /\Mn@UseConfigOrDefault{sub:series}{#4}{#4}%
1039     /\Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
1040         \Mn@UseConfigOrDefault{sub:shape}{#5}{#5}}%
1041     }{}%
1042 \fi
1043 }
1044 \otf@makeglobal\Mn@DeclareFontShape}
1045 \otf@makeglobal{\string\Mn@DeclareFontShape}

```

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

```

1046 \newcommand*\Mn@DeclareLargeFontFamily[3][LF,OsF,TLF,TOf]{%
1047     \Mn@DeclareFontFamily{#1}{#2}{#3}
1048     {m, sb, b, bx, eb} {n, it, sc, ssc, scit, sscit, sw, scsl, scsw, sscsl, sscsw, sl}%
1049 }
1050 \newcommand*\Mn@DeclareSmallFontFamily[3][LF,OsF,TLF,TOf]{%
1051     \Mn@DeclareFontFamily{#1}{#2}{#3}
1052     {m, sb, b, bx, eb} {n, it, sl}%
1053 }
1054 \newcommand*\Mn@DeclareMathFontFamily[3][TOf]{%
1055     \Mn@DeclareFontFamily[\skewchar\font=255]{#1}{#2}{#3}
1056     {m, sb, b, bx, eb} {n, it}%
1057 }

```

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

```

1058 \otf@makeglobal{Mn@DeclareLargeFontFamily}
1059 \otf@makeglobal{\string\Mn@DeclareLargeFontFamily}
1060 \otf@makeglobal{Mn@DeclareSmallFontFamily}
1061 \otf@makeglobal{\string\Mn@DeclareSmallFontFamily}
1062 \otf@makeglobal{Mn@DeclareMathFontFamily}
1063 \otf@makeglobal{\string\Mn@DeclareMathFontFamily}
1064 \newcommand*\Mn@DeclareFontFamily[6][\%
1065   \@for\Mn@variant:=#2\do{\%
1066     \DeclareFontFamily {#3}{#4-\Mn@variant}{#1}%
1067   }%
1068   \Mn@DeclareFontShapes{#3}{#4}
1069     {#5} {#6} {#2}%
1070 }
1071 \otf@makeglobal{Mn@DeclareFontFamily}
1072 \otf@makeglobal{\string\Mn@DeclareFontFamily}
1073 \newcommand*\Mn@DeclareFontShapes[5]{\%
1074   \@for\Mn@series:=#3\do{\%
1075     \@for\Mn@shape:=#4\do{\%
1076       \@for\Mn@variant:=#5\do{\%
1077         \Mn@DeclareFontShape{#1}{#2}{\Mn@series}{\Mn@shape}{\Mn@variant}%
1078       }%
1079     }%
1080   }%
1081 }
1082 \otf@makeglobal{Mn@DeclareFontShapes}

Adjust font dimension #1 of the current font. The function in #2 should replace the old
value in dimen \Mn@fontdimen with a new one (which may depend on other parameters
like \f@size).
1083 \newdimen\Mn@fontdimen
1084 \newcommand*\Mn@adjust@fontdimen[2]{\%
1085   \Mn@fontdimen=\fontdimen#1\font
1086   #2%
1087   \fontdimen#1\font=\Mn@fontdimen
1088 }
1089 \otf@makeglobal{Mn@adjust@fontdimen}
1090 \ifx\@nodocument\relax
1091   \endgroup
1092 \fi

1093 (*debug)
1094 \newcommand\old@DeclareFontFamily{}
1095 \let\old@DeclareFontFamily\DeclareFontFamily
1096 \renewcommand\DeclareFontFamily[3]{
1097   \begingroup\escapechar'\%
1098   \edef\@tempa{\noexpand\DeclareFontFamily{#1}{#2}}%
1099   \@temptokena\expandafter{\@tempa{#3}}%
1100   \message{\the\@temptokena}%
1101   \endgroup
1102   \old@DeclareFontFamily{#1}{#2}{#3}%

```

```

1103 }
1104 \newcommand\old@DeclareFontShape{}
1105 \let\old@DeclareFontShape\DeclareFontShape
1106 \renewcommand\DeclareFontShape[6]{
1107   \begingroup\escapechar'\%
1108   \edef\@tempa{\noexpand\DeclareFontShape{#1}{#2}{#3}{#4}{#5}}%
1109   \@temptokena\expandafter{\@tempa{#6}}%
1110   \message{\the\@temptokena}%
1111   \endgroup
1112   \old@DeclareFontShape{#1}{#2}{#3}{#4}{#5}{#6}%
1113 }
1114 \</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.

```

1115 \gdef\Mn@MicroType@Aliases{%
1116   \DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}%
1117   \DeclareMicrotypeAlias{MinionPro-OsF}{MinionPro}%
1118   \DeclareMicrotypeAlias{MinionPro-TLF}{MinionPro}%
1119   \DeclareMicrotypeAlias{MinionPro-TOsF}{MinionPro}%
1120 }
1121 \ifundefined{Microtype@Hook}{%
1122   \global\let\Microtype@Hook\Mn@MicroType@Aliases
1123 }{%
1124   \g@addto@macro\Microtype@Hook{\Mn@MicroType@Aliases}%
1125 }%
1126 \@ifundefined{DeclareMicroTypeAlias}{-}{\Mn@MicroType@Aliases}%
1127 \</fontdef>

```

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

```

1128 \<*fd>
1129 \input{MinionPro-FontDef.sty}%
1130 \Uextra \Mn@DeclareSmallFontFamily[Extra]{U} {MinionPro}
1131 \LGR \Mn@DeclareSmallFontFamily {LGR}{MinionPro}
1132 \LGI \Mn@DeclareSmallFontFamily {LGI}{MinionPro}
1133 \OT1 \Mn@DeclareLargeFontFamily {OT1}{MinionPro}
1134 \T1 \Mn@DeclareLargeFontFamily {T1} {MinionPro}
1135 \LY1 \Mn@DeclareLargeFontFamily {LY1}{MinionPro}
1136 \T5 \Mn@DeclareLargeFontFamily {T5} {MinionPro}
1137 \T2A \Mn@DeclareSmallFontFamily {T2A}{MinionPro}
1138 \T2B \Mn@DeclareSmallFontFamily {T2B}{MinionPro}
1139 \T2C \Mn@DeclareSmallFontFamily {T2C}{MinionPro}
1140 \TS1 \Mn@DeclareLargeFontFamily {TS1}{MinionPro}
1141 \X2 \Mn@DeclareSmallFontFamily {X2} {MinionPro}
1142 \OT2 \Mn@DeclareSmallFontFamily {OT2}{MinionPro}
1143 \OML & tosf \Mn@DeclareMathFontFamily {OML}{MinionPro}
1144 \<*OML & (lf|osf|tlf)>
1145 \<for\Mn@variant:=LF,TLF,OsF\do{>
1146   \DeclareFontFamily{OML}{MinionPro-\Mn@variant}{\skewchar\font=255}

```

```

1147 \@for\Mn@series:=m,sb,b,bx,eb\do{%
1148 \@for\Mn@shape:=n,it\do{%
1149 \DeclareFontShape{OML}{MinionPro-\Mn@variant}{\Mn@series}{\Mn@shape}%
1150 { <-> ssub*MinionPro-T0sF/\Mn@series/\Mn@shape }{}
1151 }%
1152 }%
1153 }%
1154 </OML & (lf | osf | tlf)>
1155 </fd>

```