

MinionPro Support for L^AT_EX

Achim Blumensath Andreas Bühmann Michael Zedler

v2.1 – 2007/03/15

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	11
10.3	Font selection	13
10.4	Greek letters	13
10.5	pdfT _E X to-unicode support	15
10.6	Superior and inferior figures	17
10.7	Additional symbols	20
10.8	Integral symbols	21
10.9	Open G support	22
10.10	Logos	22
10.11	AMS	23
11	Support for character protrusion	23
12	Font definition files	28

1 Overview

The MinionPro package provides support for the MinionPro font family from Adobe. You can use these fonts in a L^AT_EX 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`, 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 `dcolumn` 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, `dcolumn` 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>nonormalsize</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: <i>ABC</i>
<code>cmsy</code>	take the calligraphic symbols from Computer Modern: <i>ABC</i>
<code>swash</code>	use the swash capitals from MinionPro: <i>ȦBC</i>
<code>abx</code>	use the calligraphic symbols provided by mathabx: <i>ABCabc</i> (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: NZQRC
<code>fourierbb</code>	use the Fourier blackboard font: NZQRC
<code>lucidabb</code>	use the (commercial) Lucida Math blackboard font: NZQRC

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>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^{6,9}</code> instead of <code>example^{6,9}</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
<code>tabular, tab</code>	tabular figures
<code>proportional, prop</code>	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 L^AT_EX document classes do not support fonts with several figure versions. Therefore we provide a package `tabfigures` that patches some common document classes and packages (the standard L^AT_EX classes, KOMA-Script, `memoir`, and `amsmath`) to use tabular figures at some places. The `tabfigures` package supports the following options:

toc	use tabular figures in the table of contents
eqno	use tabular figures for equation numbers
enum	use tabular figures in enumerate environments
bib	use tabular figures in the bibliography
lineno	use tabular figures for line numbers (this affects only the doc class)

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`.

```
sc THIS IS A SAMPLE TEXT
ssc THIS IS A SAMPLE TEXT
sw This is a Sample Text
```

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

```
ƒ \digamma      κ \varkappa      β \varbeta
ε \backepsilon  3 \varbackepsilon  ħ \hbar
ħ \hslash       λ \lambdabar    λ \lambdaslash
j \jmath        ð \eth          ℔ \Bbbk
ø \slashedzero  g \openg
```

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

```
\smallfrac{⟨numerator⟩}{⟨denominator⟩}   $\frac{1}{3} \frac{5}{17}$ 
\slantfrac{⟨numerator⟩}{⟨denominator⟩}   $\frac{1}{3} \frac{5}{17}$ 
```

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

Ornaments can be accessed via the `pi` font 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	⌘	113	⌘	126	➤	139	◁
101	⌘	114	⌘	127	◁	140	▷
102	⌘	115	⌘	128	➤	141	*
103	⌘	116	⌘	129	◁	142	*
104	⌘	117	⌘	130	➤	143	*
105	⌘	118	⌘	131	◁	144	⊙
106	⌘	119	⌘	132	➤	145	□
107	⌘	120	⌘	133	◁	146	■
108	⌘	121	⌘	134	➤	147	◆
109	⌘	122	⌘	135	◁	148	✓
110	⌘	123	⌘	136	➤	149	□
111	⌘	124	⌘	137	◁	150	☑
112	⌘	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 (Ibycus transliteration scheme)

In order to typeset Greek text with the Ibycus 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 glyphtounicode
\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 ¹ , sc, scit (scsl, scsw), ssc, sscit (sscsl, 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:

¹via 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 φ instead of ϕ
- fixed `\P` in LGI encoding
- added punctuation support in LGI encoding (thanks to Jens Boerstinghaus)
- added symbols `\hslash`, `\lambdabar`, `\lambdash`
- fixed side bearings of σ 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

10 The main style file

10.1 Options

```

1 \*style}
2 \newif\if@Mn@Text@
3 \newif\if@Mn@Math@
4 \@Mn@Text@true
5 \@Mn@Math@true
6 \DeclareOption{onlytext}{\@Mn@Text@true\@Mn@Math@false}
7 \DeclareOption{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; only MinionPro integrals are handled here in `MinionPro`.

```

8 \newcommand\Mn@minionint@opticals{-NoOpticals}
9 \newcommand\Mn@minionint@bold{-Bold}
10 \DeclareOption{slides}{%
11   \def\Mn@minionint@opticals{-NoOpticals}%
12   \PassOptionsToPackage{slides}{MinionPro-FontDef}}
13 \DeclareOption{noopticals}{%
14   \def\Mn@minionint@opticals{-NoOpticals}%

```



```

15 \PassOptionsToPackage{noopticals}{MinionPro-FontDef}}
16 \DeclareOption{opticals}{%
17   \def\Mn@minionint@opticals{%
18     \PassOptionsToPackage{opticals}{MinionPro-FontDef}}
19 \DeclareOption{smallfamily}{%
20   \def\Mn@minionint@bold{-Bold}%
21   \PassOptionsToPackage{smallfamily}{MinionPro-FontDef}}
22 \DeclareOption{medfamily}{%
23   \def\Mn@minionint@bold{-Semibold}%
24   \PassOptionsToPackage{medfamily}{MinionPro-FontDef}}
25 \DeclareOption{fullfamily}{%
26   \def\Mn@minionint@bold{-Semibold}%
27   \PassOptionsToPackage{fullfamily}{MinionPro-FontDef}}
28 \DeclareOption{normalsize}{%
29   \PassOptionsToPackage{normalsize}{MinionPro-FontDef}}
30 \DeclareOption{nonnormalsize}{%
31   \PassOptionsToPackage{nonnormalsize}{MinionPro-FontDef}}

```

Figure style

```

32 \newcommand\Mn@Text@Fig{OsF}
33 \newcommand\Mn@Math@Fig{OsF}
34 \newcommand\Mn@Text@Family{MinionPro-\Mn@Text@Fig}
35 \newcommand\Mn@Math@Family{MinionPro-\Mn@Math@Fig}
36 \newcommand\Mn@Math@TFamily{MinionPro-T\Mn@Math@Fig}
37 \newcommand\Mn@Math@LetterShape{it}
38 \DeclareOption{textosf}{\def\Mn@Text@Fig{OsF}}
39 \DeclareOption{textlf}{\def\Mn@Text@Fig{LF}}
40 \DeclareOption{mathosf}{\def\Mn@Math@Fig{OsF}}
41 \DeclareOption{mathlf}{\def\Mn@Math@Fig{LF}}
42 \DeclareOption{osf}{\ExecuteOptions{textosf,mathosf}}
43 \DeclareOption{lf}{\ExecuteOptions{textlf,mathlf}}
44 \DeclareOption{mathtabular}{\let\Mn@Math@Family\Mn@Math@TFamily}

```

Calligraphic fonts

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

```

45 \newcommand\Mn@load@cal{}
46 \newcommand\Mn@load@bb{}
47 \newcommand\Mn@load@frak{}

```

Most options are handled by MnSymbol.

```

48 \DeclareOption{mnsy}{
49   \PassOptionsToPackage{mnsy}{MnSymbol}
50   \def\Mn@load@cal{
51     \SetMathAlphabet\mathcal{boldtabular}{OMS}{MnSymbolS}{b}{n}
52   }
53 }
54 \DeclareOption{cmsy}{
55   \PassOptionsToPackage{cmsy}{MnSymbol}

```

```

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

```

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.

```

73 \newcommand\Mn@load@greek{\Mn@greek@Mixed}
74 \DeclareOption{frenchmath}{%
75   \def\Mn@load@greek{\Mn@greek@Upright}%
76   \def\Mn@Math@LetterShape{n}}
77 \DeclareOption{mixedgreek}{%
78   \def\Mn@load@greek{\Mn@greek@Mixed}}
79 \DeclareOption{italicgreek}{%
80   \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}`).

```

81 \newcommand\Mn@load@amsbb{
82   \let\mathbb\undefined
83   \let\Bbbk\undefined
84   \DeclareMathAlphabet\mathbb{U}{msb}{m}{n}
85   \newcommand\Bbbk{\mathbb{\mathchar"717C}}
86 \newcommand\Mn@load@lucidabb{
87   \let\mathbb\undefined
88   \let\Bbbk\undefined
89   \DeclareFontFamily{U}{h1cm}{}
90   \DeclareFontShape{U}{h1cm}{m}{n}{<->s*[0.92] h1cra }{}
91   \DeclareMathAlphabet\mathbb{U}{h1cm}{m}{n}
92   \newcommand\Bbbk{\mathbb{k}}

```

```

93 \newcommand\Mn@load@fourierbb{
94   \let\mathbb\undefined
95   \let\Bbbk\undefined
96   \DeclareFontFamily{U}{futm}{}
97   \DeclareFontShape{U}{futm}{m}{n}{<->s*[0.95] fourier-bb }{}
98   \DeclareMathAlphabet\mathbb{U}{futm}{m}{n}
99   \newcommand\Bbbk{\mathbb{k}}
100 \DeclareOption{amsbb}  {\let\Mn@load@bb\Mn@load@amsbb}
101 \DeclareOption{lucidabb} {\let\Mn@load@bb\Mn@load@lucidabb}
102 \DeclareOption{fourierbb} {\let\Mn@load@bb\Mn@load@fourierbb}

```

Integrals

```

103 \newcommand\Mn@load@integrals{}
104 \DeclareOption{minionint}{\def\Mn@load@integrals{\Mn@Decl@Minion@Ints}}

```

Miscellaneous options

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

```

105 \DeclareOption{footnotefigures}{%
106   \def\@makefnmark{%
107     \begingroup
108     \normalfont
109     \fontfamily{MinionPro-Extra}\fontencoding{U}\selectfont
110     \@thefnmark
111     \endgroup}}
112 %
113 \newcommand\Mn@Define@Open@g{}
114 \DeclareOption{openg}{%
115   \def\Mn@Define@Open@g{%
116     \mathcode'g="8000%
117     \DeclareMathSymbol{\Mn@g}{\mathalpha}{letters}'g}%
118     \begingroup
119     \lccode'\~='\g
120     \lowercase{\gdef~{\ifnum\the\mathgroup=\m@ne \openg \else \Mn@g \fi}}%
121     \endgroup
122   }}
123 %
124 \newcommand\Mn@Quote@Spacing{}
125 \DeclareOption{loosequotes}{%
126   \def\Mn@Quote@Spacing{\Mn@Quote@Spacing@Loose}}

```

Defaults

```

127 \ExecuteOptions{amsbb,eufrak}
128 \ProcessOptions\relax

```

10.2 Font declarations

```

129 \RequirePackage{MinionPro-FontDef}
130 \ifpackageloaded{textcomp}{\RequirePackage{textcomp}}

```

```

131
132 \if@Mn@Math@
133 \RequirePackage{MnSymbol}[2007/01/21 v1.4]

```

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

```

134 \ifundefined{mathfrak}{%
135 \RequirePackage{eufrak}%
136 \SetMathAlphabet\EuFrak{boldtabular}{U}{euf}{b}{n}}{}
137 \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).

```

138 \if@Mn@Text@
139 \edef\rmdefault{\Mn@Text@Family}
140 \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.

```

141 \def\Mn@Quote@Spacing@Loose{%
142 \ifpackageloaded{microtype}{\RequirePackage[kerning=true]{microtype}}
143 \ifundefined{SetExtraKerning}{}{
144 \let\Mn@Set@Quote@Spacing\SetExtraKerning}
145 % \SetExtraKerning
146 % [ unit = 1em ]
147 % { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2},
148 % family = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
149 % shape = n }
150 % { \textquotedblleft = {30,30}, \textquotedblright = {30,30},
151 % \textquoteleft = {30,30}, \textquoteright = {30,30} }
152 }
153 \newcommand*\Mn@Set@Quote@Spacing[3]{}{}
154 \Mn@Quote@Spacing
155 \Mn@Set@Quote@Spacing
156 [ unit = 1em ]
157 { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2},
158 family = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
159 shape = {n,it} }
160 { \textquotedblleft = {30,30}, \textquotedblright = {30,30},
161 \textquoteleft = {30,30}, \textquoteright = {30,30} }
162 \fi

```

Math fonts

Redefine the standard math versions normal and bold.

```

163 \if@Mn@Math@
164 \DeclareSymbolFont{operators} {T1} {\Mn@Math@Family}{m}{n}
165 \DeclareSymbolFont{letters} {OML}{MinionPro-T0sF}{m}{\Mn@Math@LetterShape}
166 \SetSymbolFont{operators}{bold}{T1} {\Mn@Math@Family}{eb}{n}
167 \SetSymbolFont{letters} {bold}{OML}{MinionPro-T0sF}{eb}{\Mn@Math@LetterShape}
168 \DeclareMathAlphabet\mathbf {T1} {\Mn@Math@Family}{eb}{n}
169 \DeclareMathAlphabet\mathit {T1} {\Mn@Math@Family}{m}{it}
170 \SetMathAlphabet\mathit {bold}{T1} {\Mn@Math@Family}{eb}{it}

```

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

```

171 \DeclareMathVersion{tabular}
172 \SetSymbolFont{operators}{tabular}{T1}{\Mn@Math@TFamily}{m}{n}
173 \SetSymbolFont{letters}{tabular}{OML}{MinionPro-T0sF}{m}{\Mn@Math@LetterShape}
174 \SetMathAlphabet\mathit{tabular}{T1}{\Mn@Math@TFamily}{m}{it}
175
176 \DeclareMathVersion{bolddtabular}
177 \SetSymbolFont{operators}{bolddtabular}{T1}{\Mn@Math@TFamily}{eb}{n}
178 \SetSymbolFont{letters}{bolddtabular}{OML}{MinionPro-T0sF}{eb}{\Mn@Math@LetterShape}
179 \SetMathAlphabet\mathit{bolddtabular}{T1}{\Mn@Math@TFamily}{eb}{it}

180 \DeclareMathAccent{\grave}{\mathalpha}{operators}{0}
181 \DeclareMathAccent{\acute}{\mathalpha}{operators}{1}
182 \DeclareMathAccent{\hat}{\mathalpha}{operators}{2}
183 \DeclareMathAccent{\tilde}{\mathalpha}{operators}{3}
184 \DeclareMathAccent{\ddot}{\mathalpha}{operators}{4}
185 \DeclareMathAccent{\mathring}{\mathalpha}{operators}{6}
186 \DeclareMathAccent{\check}{\mathalpha}{operators}{7}
187 \DeclareMathAccent{\breve}{\mathalpha}{operators}{8}
188 \DeclareMathAccent{\bar}{\mathalpha}{operators}{9}
189 \DeclareMathAccent{\dot}{\mathalpha}{operators}{10}

```

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

```

190 \Mn@load@bb
191 \Mn@load@frak
192 \Mn@load@cal
193 \fi

```

10.3 Font selection

The font selection commands such as `\figureversion`, `\textsw`, and `\textssc` are provided by the companion package `fontaxes`, which may be useful for other font families as well.

```

194 \RequirePackage{fontaxes}[2005/05/04]

```

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

```

195 \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.

```

196 \if@Mn@Math@
197 \newcommand\Mn@greek@list@upper{}

```

```

198 \newcommand\Mn@greek@list@lower{}
199 \let\Mn@greek@list@upper@gobble
200 \let\Mn@greek@list@lower@gobble

```

This macro holds one of the two list names.

```

201 \newcommand\Mn@greek@list{}
202 \newcommand*\Mn@greek@letter[3]{%
203   \expandafter\DeclareMathSymbol
204   \expandafter{\csname it#1\endcsname}{\mathord}{letters}{#2}%
205   \expandafter\DeclareMathSymbol
206   \expandafter{\csname up#1\endcsname}{\mathord}{letters}{#3}%
207   \edef\@tempa{\@car#1\@nil}%
208   \edef\Mn@greek@list{\expandafter\noexpand\csname
209     Mn@greek@list@ifnum\uccode\@tempa=\@tempa upper\else lower\fi\endcsname}%
210   \expandafter\edef\Mn@greek@list{\Mn@greek@list,#1}%
211 }

```

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

```

212 \Mn@greek@letter{\Gamma}      {'000}{ '200}
213 \Mn@greek@letter{\Delta}     {'001}{ '201}
214 \Mn@greek@letter{\Theta}     {'002}{ '202}
215 \Mn@greek@letter{\Lambda}    {'003}{ '203}
216 \Mn@greek@letter{\Xi}       {'004}{ '204}
217 \Mn@greek@letter{\Pi}       {'005}{ '205}
218 \Mn@greek@letter{\Sigma}    {'006}{ '206}
219 \Mn@greek@letter{\Upsilon}   {'007}{ '207}
220 \Mn@greek@letter{\Phi}      {'010}{ '210}
221 \Mn@greek@letter{\Psi}      {'011}{ '211}
222 \Mn@greek@letter{\Omega}    {'012}{ '212}
223 \Mn@greek@letter{\alpha}    {'013}{ '213}
224 \Mn@greek@letter{\beta}     {'014}{ '214}
225 \Mn@greek@letter{\gamma}    {'015}{ '215}
226 \Mn@greek@letter{\delta}   {'016}{ '216}
227 \Mn@greek@letter{\epsilon}  {'017}{ '217}
228 \Mn@greek@letter{\zeta}     {'020}{ '220}
229 \Mn@greek@letter{\eta}      {'021}{ '221}
230 \Mn@greek@letter{\theta}    {'022}{ '222}
231 \Mn@greek@letter{\iota}     {'023}{ '223}
232 \Mn@greek@letter{\kappa}    {'024}{ '224}
233 \Mn@greek@letter{\lambda}   {'025}{ '225}
234 \Mn@greek@letter{\mu}       {'026}{ '226}
235 \Mn@greek@letter{\nu}       {'027}{ '227}
236 \Mn@greek@letter{\xi}       {'030}{ '230}
237 \Mn@greek@letter{\pi}       {'031}{ '231}
238 \Mn@greek@letter{\rho}      {'032}{ '232}
239 \Mn@greek@letter{\sigma}    {'033}{ '233}
240 \Mn@greek@letter{\tau}      {'034}{ '234}
241 \Mn@greek@letter{\upsilon}  {'035}{ '235}
242 \Mn@greek@letter{\phi}      {'036}{ '236}
243 \Mn@greek@letter{\chi}      {'037}{ '237}
244 \Mn@greek@letter{\psi}      {'040}{ '240}

```

```

245 \Mn@greek@letter{\omega}      {'041}{'241}
246 \Mn@greek@letter{\varepsilon} {'042}{'242}
247 \Mn@greek@letter{\vartheta}   {'043}{'243}
248 \Mn@greek@letter{\varpi}      {'044}{'244}
249 \Mn@greek@letter{\varrho}     {'045}{'245}
250 \Mn@greek@letter{\varsigma}   {'046}{'246}
251 \Mn@greek@letter{\varphi}     {'047}{'247}

```

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

```

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

```

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

```

258 \newcommand*\Mn@greek@select[2]{%
259   \expandafter\let\expandafter\Mn@greek@list\csname Mn@greek@list@#2\endcsname
260   \for\@tempa:=\Mn@greek@list\do{%
261     \expandafter\let\csname\@tempa\expandafter\endcsname
262     \csname#1\@tempa\endcsname
263   }%
264 }
265 \newcommand*\Mn@greek@Upright{%
266   \Mn@greek@select{up}{upper}%
267   \Mn@greek@select{up}{lower}%
268 }
269 \newcommand*\Mn@greek@Italic{%
270   \Mn@greek@select{it}{upper}%
271   \Mn@greek@select{it}{lower}%
272 }
273 \newcommand*\Mn@greek@Mixed{%
274   \Mn@greek@select{up}{upper}%
275   \Mn@greek@select{it}{lower}%
276 }

```

Finally initialise the Greek letters.

```

277 \Mn@load@greek
278 \fi

```

10.5 pdf_{TEX} to-unicode support

Old versions of MinionPro have non-standard glyph names.

```

279 \ifundefined{pdfglyphtounicode}{}{
280   \pdfglyphtounicode{uniEFD5}{03DD}% uni03DD
281   \pdfglyphtounicode{uniEFED}{02D9}% dotaccent.cap
282   \pdfglyphtounicode{uniEFEE}{02D8}% breve.cap
283   \pdfglyphtounicode{uniEFF1}{02DB}% ogonek.cap
284   \pdfglyphtounicode{uniEFF2}{00B8}% cedilla.cap

```

285 \pdfglyphtounicode{uniEFF3}{02DA}% ring.cap
286 \pdfglyphtounicode{uniEFF5}{02DC}% tilde.cap
287 \pdfglyphtounicode{uniEFF7}{02C6}% circumflex.cap
288 \pdfglyphtounicode{uniF628}{2030}% perthousand.oldstyle
289 \pdfglyphtounicode{uniF62C}{0028}% parenleft.denominator
290 \pdfglyphtounicode{uniF62D}{0029}% parenright.denominator
291 \pdfglyphtounicode{uniF631}{0028}% parenleft.numerator
292 \pdfglyphtounicode{uniF632}{0029}% parenright.numerator
293 \pdfglyphtounicode{uniF638}{0030}% zero.slash
294 \pdfglyphtounicode{uniF639}{0030}% zero.fitted
295 \pdfglyphtounicode{uniF63A}{0032}% two.fitted
296 \pdfglyphtounicode{uniF63B}{0033}% three.fitted
297 \pdfglyphtounicode{uniF63C}{0034}% four.fitted
298 \pdfglyphtounicode{uniF63D}{0035}% five.fitted
299 \pdfglyphtounicode{uniF63E}{0036}% six.fitted
300 \pdfglyphtounicode{uniF63F}{0037}% seven.fitted
301 \pdfglyphtounicode{uniF640}{0038}% eight.fitted
302 \pdfglyphtounicode{uniF641}{0039}% nine.fitted
303 \pdfglyphtounicode{uniF642}{0025}% percent.oldstyle
304 \pdfglyphtounicode{uniF643}{0030}% zero.taboldstyle
305 \pdfglyphtounicode{uniF644}{0031}% one.taboldstyle
306 \pdfglyphtounicode{uniF645}{0032}% two.taboldstyle
307 \pdfglyphtounicode{uniF646}{0033}% three.taboldstyle
308 \pdfglyphtounicode{uniF647}{0034}% four.taboldstyle
309 \pdfglyphtounicode{uniF648}{0035}% five.taboldstyle
310 \pdfglyphtounicode{uniF649}{0036}% six.taboldstyle
311 \pdfglyphtounicode{uniF64A}{0037}% seven.taboldstyle
312 \pdfglyphtounicode{uniF64B}{0038}% eight.taboldstyle
313 \pdfglyphtounicode{uniF64C}{0039}% nine.taboldstyle
314 \pdfglyphtounicode{uniF64D}{20A1}% colonmonetary.taboldstyle
315 \pdfglyphtounicode{uniF64E}{20AC}% Euro.taboldstyle
316 \pdfglyphtounicode{uniF64F}{0192}% florin.taboldstyle
317 \pdfglyphtounicode{uniF650}{0023}% numbersign.taboldstyle
318 \pdfglyphtounicode{uniF651}{00A3}% sterling.taboldstyle
319 \pdfglyphtounicode{uniF652}{00A5}% yen.taboldstyle
320 \pdfglyphtounicode{uniF653}{0024}% dollar.taboldstyle
321 \pdfglyphtounicode{uniF654}{00A2}% cent.taboldstyle
322 \pdfglyphtounicode{uniF655}{0030}% zero.denominator
323 \pdfglyphtounicode{uniF656}{0031}% one.denominator
324 \pdfglyphtounicode{uniF657}{0032}% two.denominator
325 \pdfglyphtounicode{uniF658}{0033}% three.denominator
326 \pdfglyphtounicode{uniF659}{0034}% four.denominator
327 \pdfglyphtounicode{uniF65A}{0035}% five.denominator
328 \pdfglyphtounicode{uniF65B}{0036}% six.denominator
329 \pdfglyphtounicode{uniF65C}{0037}% seven.denominator
330 \pdfglyphtounicode{uniF65D}{0038}% eight.denominator
331 \pdfglyphtounicode{uniF65E}{0039}% nine.denominator
332 \pdfglyphtounicode{uniF65F}{002C}% comma.denominator
333 \pdfglyphtounicode{uniF660}{002E}% period.denominator
334 \pdfglyphtounicode{uniF661}{0030}% zero.numerator


```

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

```

10.6 Superior and inferior figures

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

```

377 \def\@for@tok#1:=#2\do#3{%
378 \expandafter\def\expandafter\@fortmp\expandafter{#2}%
379 \ifx\@fortmp\@empty \else

```

```

380 \expandafter\@forloop@tok#2\@nil\@nil\@#1{#3}%
381 \fi}
382 \def\@forloop@tok#1#2#3\@#4#5{%
383 \def#4{#1}%
384 \ifx #4\@nnil \else
385 #5%
386 \def#4{#2}%
387 \ifx #4\@nnil \else
388 #5\@iforloop@tok #3\@#4{#5}%
389 \fi\fi}
390 \def\@iforloop@tok#1#2\@#3#4{%
391 \def#3{#1}%
392 \ifx #3\@nnil
393 \expandafter\@forloop
394 \else
395 #4\relax\expandafter\@iforloop@tok
396 \fi
397 #2\@#3{#4}}
398 %
399 \newcommand*\Mn@extra@font{%
400 \fontencoding{U}\fontfamily{MinionPro-Extra}\selectfont}
401 \newcommand*\@numerator@fig[1]{\Mn@extra@font\@@numerator@fig{#1}}
402 \newcommand*\@denominator@fig[1]{\Mn@extra@font\@@denominator@fig{#1}}
403 \newcommand*\@superior@fig[1]{\Mn@extra@font\@@superior@fig{#1}}
404 \newcommand*\@inferior@fig[1]{\Mn@extra@font\@@inferior@fig{#1}}
405 \newcommand*\@@numerator@fig[1]{%
406 \@for@tok\@nf@fig:=#1\do{%
407 \ifcase\@nf@fig
408 \char'00%
409 \or\char'01%
410 \or\char'02%
411 \or\char'03%
412 \or\char'04%
413 \or\char'05%
414 \or\char'06%
415 \or\char'07%
416 \or\char'10%
417 \or\char'11%
418 \else
419 \@latex@error{invalid argument to \string\@@numerator@fig}%
420 \fi
421 }}
422 \newcommand*\@@denominator@fig[1]{%
423 \@for@tok\@nf@fig:=#1\do{%
424 \ifcase\@nf@fig
425 \char'20%
426 \or\char'21%
427 \or\char'22%
428 \or\char'23%
429 \or\char'24%

```

```

430 \or\char'25%
431 \or\char'26%
432 \or\char'27%
433 \or\char'30%
434 \or\char'31%
435 \else
436 \latexerror{invalid argument to \string\@denominator@fig}%
437 \fi
438 }}
439 \newcommand*\@superior@fig[1]{%
440 \for@tok\nf@fig:=#1\do{%
441 \ifcase\nf@fig
442 \char'60%
443 \or\char'61%
444 \or\char'62%
445 \or\char'63%
446 \or\char'64%
447 \or\char'65%
448 \or\char'66%
449 \or\char'67%
450 \or\char'70%
451 \or\char'71%
452 \else
453 \latexerror{invalid argument to \string\@superior@fig}%
454 \fi
455 }}
456 \newcommand*\@inferior@fig[1]{%
457 \for@tok\nf@fig:=#1\do{%
458 \ifcase\nf@fig
459 \char'100%
460 \or\char'101%
461 \or\char'102%
462 \or\char'103%
463 \or\char'104%
464 \or\char'105%
465 \or\char'106%
466 \or\char'107%
467 \or\char'110%
468 \or\char'111%
469 \else
470 \latexerror{invalid argument to \string\@inferior@fig}%
471 \fi
472 }}
\ensure@text switches to text mode, if necessary.
473 \newcommand*\ensure@text[1]{%
474 \ifmmode
475 \Mn@Text@With@MathVersion{#1}%
476 \else
477 #1%

```

```

478 \fi}
\smallfrac and \slantfrac assemble numerical fractions.
479 \newcommand*\@smallfrac[2]{%
480 \leavevmode
481 \setbox\@tempboxa
482 \vbox{%
483 \baselineskip\z@skip%
484 \lineskip.25ex%
485 \lineskiplimit-\maxdimen
486 \ialign{\hfil##\hfil\crcr
487 \vbox to 2.13ex{\vss\hbox{\@numerator@fig{#1}}\vskip.68ex}\crcr
488 \leavevmode\leaders\hrule height 1.1ex depth -1.01ex\hfill\crcr
489 \vtop to 1ex{\vbox{\hbox{\@denominator@fig{#2}}\vss}\crcr
490 \noalign{\vskip-1.47ex}}}%
491 \dp\@tempboxa=0.49ex%
492 \box\@tempboxa}
493 \newcommand*\@slantfrac[2]{%
494 {\Mn@extra@font\@numerator@fig{#1}\kern-.05em/\kern-.06em\@denominator@fig{#2}}
495 \DeclareRobustCommand*\smallfrac[2]{\ensure@text{\kern0.06em\@smallfrac{#1}{#2}\kern0.09em}}
496 \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.

```

497 \if@Mn@Math@
498 \let\hbar\undefined
499 \DeclareMathSymbol{\hbar} {\mathord}{letters}'{265}
500 \DeclareMathSymbol{\uphbar} {\mathord}{letters}'{255}
501 \DeclareMathSymbol{\partial} {\mathord}{letters}'{100}
502 \DeclareMathSymbol{\uppartial} {\mathord}{letters}'{300}
503 \DeclareMathSymbol{\ell} {\mathord}{letters}'{140}
504 \DeclareMathSymbol{\upe11} {\mathord}{letters}'{340}
505 \DeclareMathSymbol{\slashedzero} {\mathord}{letters}'{257}
506 \DeclareMathSymbol{\upimath} {\mathord}{letters}'{373}
507 \DeclareMathSymbol{\upjmath} {\mathord}{letters}'{374}
508 \DeclareMathSymbol{\varsmallint} {\mathord}{letters}'{376}
509 \DeclareMathSymbol{\openg} {\mathalpha}{letters}'{267}
510 \DeclareRobustCommand\lambdabar {\middleslash\lambda}
511 \DeclareRobustCommand\lambdaslash {\middleslash\lambda}
512 \fi

```

Archaic Greek letters not provided by MinionPro.

```

513 \if@Mn@Text@
514 %\def\Qoppa{\reflectbox{P}}
515 %\def\Sampi{\begingroup\fontfamily{cmr}\fontencoding{LGR}\selectfont\char23\endgroup}
516 \let\Stigma\stigma
517
518 % fix \r A
519 \DeclareTextCompositeCommand{\r}{OT1}{A}

```

```

520     {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
521     \oalign{\hss\raise.67\dimen@\hbox{\char23}\hss\crrc A}}
522
523 \DeclareEncodingSubset{TS1}{MinionPro-LF} {1}%
524 \DeclareEncodingSubset{TS1}{MinionPro-TLF} {1}%
525 \DeclareEncodingSubset{TS1}{MinionPro-OsF} {1}%
526 \DeclareEncodingSubset{TS1}{MinionPro-TOsF}{1}%
527 \AtBeginDocument{
528   \UndeclareTextCommand{\textvisiblespace}{T1}%
529   \UndeclareTextCommand{\textcompwordmark}{T1}%
530   \UndeclareTextCommand{\textsterling}{T1}%
531   \UndeclareTextCommand{\j}{T1}%
532   \UndeclareTextCommand{\j}{LY1}%
533 }
534 \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.

```

535 \if@Mn@Math@
536 \newcommand\Mn@Decl@Minion@Ints{%

```

Replace MnSymbolF by MnSymbolFI.

```

537 \DeclareFontFamily{U}{MnSymbolFI}{}
538 \DeclareFontShape{U}{MnSymbolFI}{m}{it}{
539   <-6> MnSymbolFI\Mn@minionint@opticals5
540   <6-7> MnSymbolFI\Mn@minionint@opticals6
541   <7-8> MnSymbolFI\Mn@minionint@opticals7
542   <8-9> MnSymbolFI\Mn@minionint@opticals8
543   <9-10> MnSymbolFI\Mn@minionint@opticals9
544   <10-12> MnSymbolFI\Mn@minionint@opticals10
545   <12-> MnSymbolFI\Mn@minionint@opticals12
546 }{}
547 \DeclareFontShape{U}{MnSymbolFI}{b}{it}{
548   <-6> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals5
549   <6-7> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals6
550   <7-8> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals7
551   <8-9> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals8
552   <9-10> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals9
553   <10-12> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals10
554   <12-> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals12
555 }{}
556 \DeclareSymbolFont{symbols} {U}{MnSymbolFI}{m}{it}
557 \SetSymbolFont{symbols}{bold}{U}{MnSymbolFI}{b}{it}

```

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

```

558 \let\varint\tint
559 \let\variint\tiint

```

```

560 \let\variint\tiint
561 \let\variiint\tiiiint
562 \let\varidotsint\tidotsint
563 \let\varlandupint\tlandupint
564 \let\varlanddownint\tlanddownint
565 \let\varstrokedint\tstrokedint
566 \let\varoint\toint
567 \let\varoiint\toiint
568 \let\varrcirclearightint\trcirclearightint
569 \let\varlcirclearightint\tlcirclearightint
570 \let\varrcircleleftint\trcircleleftint
571 \let\varlcircleleftint\tlcircleleftint
572 \let\varsumint\tsumint

```

Replace the symbols with the new integrals.

```

573 \DeclareMathSymbol\tint \mathop{symbols}{112}
574 \DeclareMathSymbol\tiint \mathop{symbols}{114}
575 \DeclareMathSymbol\tiiiint \mathop{symbols}{116}
576 \DeclareMathSymbol\tiiiint \mathop{symbols}{118}
577 \DeclareMathSymbol\tidotsint \mathop{symbols}{120}
578 \DeclareMathSymbol\tlandupint \mathop{symbols}{122}
579 \DeclareMathSymbol\tlanddownint \mathop{symbols}{124}
580 \DeclareMathSymbol\tstrokedint \mathop{symbols}{126}
581 \DeclareMathSymbol\toint \mathop{symbols}{128}
582 \DeclareMathSymbol\toiint \mathop{symbols}{130}
583 \DeclareMathSymbol\trcirclearightint \mathop{symbols}{132}
584 \DeclareMathSymbol\tlcirclearightint \mathop{symbols}{134}
585 \DeclareMathSymbol\trcircleleftint \mathop{symbols}{136}
586 \DeclareMathSymbol\tlcircleleftint \mathop{symbols}{138}
587 \DeclareMathSymbol\tsumint \mathop{symbols}{140}
588 \let\intop\tint
589 \let\ointop\toint
590 }
591 \Mn@load@integrals
592 \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.

```

593 \if@Mn@Math@
594 \Mn@Define@Open@g
595 \fi

```

10.10 Logos

Correct logos.

```

596 \if@Mn@Text@
597 \def\TeX{T\kern-.1667em\lower.4ex\hbox{E}\kern-.125emX\@}

```

```

598 \DeclareRobustCommand{\LaTeX}{L\kern-.32em%
599     {\sbox\z@ T%
600       \vbox to\ht\z@{\hbox{\check@mathfonts
601         \fontsize\sf@size\z@
602         \math@fontsfalse\selectfont
603         A}%
604         \vss}%
605     }%
606     \kern-.15em%
607     \TeX}
608 \fi

```

10.11 AMS

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

```

609 \def\macc@set@skewchar#1{%
610   \begingroup
611   \ifnum\mathgroup=\m@ne \let\@tempa\@ne
612   \else
613     \ifnum\skewchar\textfont\mathgroup=\m@ne \let\@tempa\@ne
614     \else \let\@tempa\mathgroup
615   \fi
616   \fi
617   \count@=\skewchar\textfont\@tempa
618   \ifnum\count@=\m@ne
619     \endgroup
620     \def\macc@skewchar{}
621   \else
622     \advance\count@"7100
623     \edef\@tempa{\endgroup
624       \mathchardef\noexpand\macc@skewchar=\number\count@\relax}%
625     \@tempa
626   \fi
627   #1%
628 }

```

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

```

629 \if@Mn@Text@
630 \normalfont
631 \fi
632 \endstyle

```

11 Support for character protrusion

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

```

633 \*mtcfg)
634 \SetProtrusion
635 [ name = MinionPro-OT1-Roman ]

```

```

636 { encoding = OT1,
637   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
638   shape    = n }
639 {
640   A = {40,40},
641   F = { ,60},
642   J = {90, },
643   K = { ,50},
644   L = { ,60},
645   T = {50,50},
646   V = {40,40},
647   W = {30,30},
648   X = {50,50},
649   Y = {50,50},
650   k = { ,60},
651   r = { ,80},
652   t = { ,100},
653   v = {70,70},
654   w = {40,40},
655   x = {60,60},
656   y = {70,70},
657   ! = {70,180},
658   ( = {60,30}, ) = {30,60},
659   [ = {100,160}, ] = {160,100},
660   {,} = {440,700},
661   . = {660,700},
662   : = {400,480},
663   ; = {350,440},
664   - = {700,700},
665   \textendash = {390,480}, \textemdash = {220,270},
666   \textquotedblleft = {380,250}, \textquotedblright = {250,380},
667   \textquoteleft = {670,450}, \textquoteright = {450,670},
668 }

669 \SetProtrusion
670 [ name = MinionPro-T1-Roman,
671   load = MinionPro-OT1-Roman ]
672 { encoding = T1,
673   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
674   shape    = n }
675 {
676   023 = { ,40}, % fft ligature
677   032 = { ,50}, % ft ligature
678   191 = {30,30}, % Th ligature
679   127 = {620,700}, % hyphen
680   \AE = {40, }, % AE
681   \quotesinglbase = {670,670}, \quotedblbase = {370,370},
682   \guilsinglleft = {500,360}, \guilsinglright = {360,500},
683   \guillemotleft = {320,230}, \guillemotright = {230,320},
684 }

```



```

685 \SetProtrusion
686 [ name = MinionPro-OT1-Italic]
687 { encoding = OT1,
688   family = {MinionPro-0sF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
689   shape = {it,s,sw} }
690 {
691   A = {120,50},
692   B = {90,-50},
693   C = {50,-60},
694   D = {70,-30},
695   E = {90,-50},
696   F = {100,-40},
697   G = {50,-60},
698   H = {70,-40},
699   I = {150,-90},
700   J = {250,-130},
701   K = {80,-50},
702   L = {90,60},
703   M = {60,-40},
704   N = {70,-40},
705   O = {70,-30},
706   P = {70,-110},
707   Q = {40,-40},
708   R = {80,-50},
709   S = {70,-70},
710   T = {130, },
711   U = {70,-40},
712   V = {120,30},
713   W = {90,20},
714   X = {50, },
715   Y = {160, },
716   Z = {50,-50},
717   d = {60,-60},
718   f = { , -190},
719   027 = { , -70}, % ff ligature
720   g = {-70,-70},
721   i = { , -110},
722   025 = { , -60}, % dotlessi
723   028 = { , -60}, % fi ligature
724   030 = { , -30}, % ffi ligature
725   j = {-90,-150},
726   p = {-40, },
727   r = { , 80},
728   t = { , 100},
729   v = {90, },
730   w = {60,10},
731   x = {90, },
732   ! = {190,40},
733   ( = {90, }, ) = {90, },
734   [ = {90,90}, ] = {120,60},

```

```

735 {,} = {210,680},
736 . = {640,680},
737 : = {380,430},
738 ; = { ,430},
739 - = {750,750},
740 \textquoteleft = {690,140}, \textquoteright = {470,230},
741 \textendash = {400,500}, \textemdash = {220,280},
742 \textquotedblleft = {520,130}, \textquotedblright = {520,130},
743 }
744 \SetProtrusion
745 [ name = MinionPro-T1-Italic,
746 load = MinionPro-OT1-Italic ]
747 { encoding = T1,
748 family = {MinionPro-0sF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
749 shape = {it,s,sw} }
750 {
751 023 = { ,40}, % fft ligature
752 032 = { ,50}, % ft ligature
753 191 = {80,30}, % Th ligature
754 127 = {660,750}, % hyphen
755 \AE = {90,-40}, % AE
756 131 = {80,-30}, % Dcaron
757 132 = {70,-40}, % Ecaron
758 156 = {80,-60}, % IJ
759 \OE = {50,-30}, % OE
760 188 = { , -80}, % ij
761 184 = {70,70}, % ydieresis
762 253 = {70,70}, % yacute
763 \quotesinglbase = {220,700}, \quotedblbase = {130,400},
764 \guilsinglleft = {500,180}, \guilsinglright = {350,350},
765 \guillemotleft = {310,110}, \guillemotright = {230,230},
766 }

```

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

```

767 \SetProtrusion
768 [ name = MinionPro-OT1-Smallcaps ]
769 { encoding = OT1,
770 family = {MinionPro-0sF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
771 shape = {sc,ssc} }
772 {}
773 \SetProtrusion
774 [ name = MinionPro-T1-Smallcaps,
775 load = MinionPro-OT1-Smallcaps ]
776 { encoding = T1,
777 family = {MinionPro-0sF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
778 shape = {sc,ssc} }
779 {}
780 \SetProtrusion

```

```

781 [ name      = MinionPro-OT1-SmallcapsItalic ]
782 { encoding = OT1,
783   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
784   shape    = {scit,sscit} }
785 {}

786 \SetProtrusion
787 [ name      = MinionPro-T1-SmallcapsItalic,
788   load      = MinionPro-OT1-SmallcapsItalic ]
789 { encoding = T1,
790   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
791   shape    = {scit,sscit} }
792 {}

793 \SetProtrusion
794 [ name      = MinionPro-other-Roman ]
795 { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
796   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
797   shape    = n }
798 {
799   ! = {70,180},
800   ( = {60,30},   ) = {30,60},
801   [ = {100,160}, ] = {160,100},
802   {,} = {440,700},
803   . = {660,700},
804   : = {400,480},
805   ; = {350,440},
806   - = {700,700},
807   \textendash      = {390,480}, \textemdash      = {220,270},
808   \textquotedblleft = {380,250}, \textquotedblright = {250,380},
809   \textquoteleft   = {670,450}, \textquoteright   = {450,670},
810 }

811 \SetProtrusion
812 [ name      = MinionPro-other-Italic ]
813 { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
814   family   = {MinionPro-OsF,MinionPro-LF,MinionPro-T0sF,MinionPro-TLF},
815   shape    = {it,sl,sw} }
816 {
817   ! = {190,40},
818   ( = {90,  },   ) = {90,  },
819   [ = {90,90},   ] = {120,60},
820   {,} = {210,680},
821   . = {640,680},
822   : = {380,430},
823   ; = {  ,430},
824   - = {750,750},
825   \textquoteleft   = {690,140}, \textquoteright   = {470,230},
826   \textendash      = {400,500}, \textemdash      = {220,280},
827   \textquotedblleft = {520,130}, \textquotedblright = {520,130},
828 }
829 \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.

```
830 (*fontdef)
831 \ifx\Mn@DeclareFontShape\undefined\else\endinput\fi
```

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

```
832 \ifx\@nodocument\relax
833   \input{otfontdef.sty}
834 \else
835   \NeedsTeXFormat{LaTeX2e}
836   \RequirePackage{otfontdef}
837 \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.

```
838 \ifx\@nodocument\relax
839   \begingroup\escapechar'\
840 \fi
```

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

```
841 \newcommand\Mn@option@opticals{noopticals}
842 \newcommand\Mn@option@fontset{smallfamily}
843 \newdimen\Mn@option@normalsize
844 \global\Mn@option@normalsize10pt
```

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

```
845 \newif\ifMn@option@normalsize
846 \Mn@option@normalsize>true

847 \ifx\@nodocument\relax\else
848   \DeclareOption{slides}    {\let\Mn@option@opticals\CurrentOption}
849   \DeclareOption{opticals}  {\let\Mn@option@opticals\CurrentOption}
850   \DeclareOption{noopticals}{\let\Mn@option@opticals\CurrentOption}
851   \DeclareOption{smallfamily}{\let\Mn@option@fontset\CurrentOption}
852   \DeclareOption{medfamily} {\let\Mn@option@fontset\CurrentOption}
853   \DeclareOption{fullfamily}{\let\Mn@option@fontset\CurrentOption}
854   \DeclareOption{normalsize}{\Mn@option@normalsize>true}
855   \DeclareOption{nonnormalsize}{\Mn@option@normalsize>false}
856   \ExecuteOptions{smallfamily,noopticals,normalsize}
857   \ProcessOptions\relax
858 \fi
```

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

```
859 \ifMn@option@normalsize
860 \begingroup
861 \def\set@fontsize#1#2#3#4\@nil{%
862   \@defaultunits\global\Mn@option@normalsize#2pt\relax\@nnil}%
863 \normalsize\@nil
864 \endgroup
865 \fi
```

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

```
866 \otf@makeglobal\Mn@option@opticals}
867 \otf@makeglobal\Mn@option@fontset}
868 \ifx\@nodocument\relax\else
869 \PackageInfo{MinionPro-FontDef}{%
870   Configuration:\space\Mn@option@fontset,\space\Mn@option@opticals,\space
871   normalsize=\the\Mn@option@normalsize}%
872 \fi
```

Configuration database

```
873 \newcount\Mn@config@cnt
874 \Mn@config@cnt=0
875 \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.

```
876 \newcommand\Mn@AddToConfig{%
877   \begingroup
878   \nfss@catcodes
879   \expandafter\endgroup
880   \Mn@AddToConfig@
881 }
882 \newcommand\Mn@AddToConfig@[3]{%
883   \advance\Mn@config@cnt\@ne
884   \@namedef{\Mn@curr@config}{#3}%
885   \otf@makeglobal{\Mn@curr@config}
886 \debug & show\expandafter\show\csname\Mn@curr@config\endcsname
887 \@for\Mn@tempa:=#2\do{%
888   \ifundefined\Mn@config@#1@\Mn@tempa}{%
889     \@temptokena{}%
890   }{%
891     \@temptokena\expandafter\expandafter\expandafter
892     {\csname Mn@config@#1@\Mn@tempa\endcsname}%
893   }%
894 \@expandtwoargs\@namedef{\Mn@config@#1@\Mn@tempa}{%
895   \the\@temptokena
896 \expandafter\noexpand\csname\Mn@curr@config\endcsname
```

```

897 }%
898 \otf@makeglobal\Mn@config@#1@\Mn@tempa}% perhaps defer to only execute once
899 (debug & show)\expandafter\show\csname Mn@config@#1@\Mn@tempa\endcsname
900 }%
901 }

```

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.

```

902 \newcommand*\Mn@UseConfig[2]{%
903 \Mn@UseConfigOrDefault{#1}{#2}}%
904 }
905 \newcommand*\Mn@UseConfigOrDefault[3]{%
906 \ifundefined\Mn@config@#1@#2}{#3}%
907 {\@nameuse\Mn@config@#1@#2}}%
908 }
909 \newcommand*\Mn@TheConfig[2]{%
910 \ifundefined\Mn@config@#1@#2}}{%
911 \expandafter\noexpand\csname Mn@config@#1@#2\endcsname
912 }%
913 }
914 \otf@makeglobal\Mn@UseConfig
915 \otf@makeglobal\Mn@UseConfigOrDefault
916 \otf@makeglobal\Mn@TheConfig

```

Here comes the configuration.

```

917 \Mn@AddToConfig{opticals}{opticals}{
918 <-8.5> otf* [optical=Capt]
919 <8.5-13.1> otf* [optical=Text]
920 <13.1-20> otf* [optical=Subh]
921 <20-> otf* [optical=Disp]
922 }
923 \Mn@AddToConfig{opticals}{noopticals}{
924 <-> otf* [optical=Text]
925 }
926 \Mn@AddToConfig{opticals}{slides}{
927 <-> otf* [optical=Capt]
928 }
929 \ifdim\Mn@option@normalsize<10.1pt
930 \Mn@AddToConfig{fontset/weight}{fullfamily/m}{
931 <-6> otf* [weight=Semibold]
932 <6-8.5> otf* [weight=Medium]

```

```

933     <8.5->    otf* [weight=Regular]
934   }
935 \else
936   \Mn@AddToConfig{fontset/weight}{fullfamily/m}{
937     <-6>    otf* [weight=Semibold]
938     <6-10.1> otf* [weight=Medium]
939     <10.1-> otf* [weight=Regular]
940   }
941 \fi
942 \Mn@AddToConfig{fontset/weight}{medfamily/m}{
943   <-6>    otf* [weight=Semibold]
944   <6->    otf* [weight=Regular]
945 }
946 \Mn@AddToConfig{fontset/weight}{smallfamily/m}{
947   <->    otf* [weight=Regular]
948 }
949 %
950 \Mn@AddToConfig{fontset/weight}{fullfamily/b,medfamily/b}{
951   <-6>    otf* [weight=Bold]
952   <6->    otf* [weight=Semibold]
953 }
954 \Mn@AddToConfig{fontset/weight}{smallfamily/b}{
955   <->    otf* [weight=Bold]
956 }
957 %
958 \Mn@AddToConfig{weight}{eb}{
959   <->    otf* [weight=Bold]
960 }
961 \Mn@AddToConfig{shape}{ssc,sscit}{
962   <->    otf* [spacing=12]
963 }
964 \Mn@AddToConfig{shape}{n,it,sw,sc,scit}{
965   <-8>    otf* [spacing=11]
966 }
967 \Mn@AddToConfig{encoding/shape}{U/n,U/it}{
968   <->    otf* [spacing=]
969 }
970 %
971 \Mn@AddToConfig{shape}{sc,ssc,scit,sscit}{
972   <->    otf* [variant=sc]
973 }
974 \Mn@AddToConfig{shape}{sw}{
975   <->    otf* [variant=swash]
976 }
977 \Mn@AddToConfig{shape}{it,scit,sscit,sw}{
978   <->    otf* MinionPro-It
979 }
980 \Mn@AddToConfig{shape}{n,sc,ssc}{
981   <->    otf* MinionPro

```

```

982 }
983 \Mn@AddToConfig{encoding/shape}{OML/it}{
984     <->     otf* [figures=] MinionPro-Mixed
985 }
986 \Mn@AddToConfig{encoding/shape}{OML/n}{
987     <->     otf* [figures=] MinionPro-French
988 }

```

Substitutions

```

989 \Mn@AddToConfig{sub:series} {sb}    {b}
990 \Mn@AddToConfig{sub:series} {bx}    {b}
991 \Mn@AddToConfig{sub:shape}  {sl}    {it}
992 \Mn@AddToConfig{sub:shape}  {scsl}  {scit}
993 \Mn@AddToConfig{sub:shape}  {sscs1} {ssc1t}
994 \Mn@AddToConfig{sub:shape}  {scsw}  {scit}
995 \Mn@AddToConfig{sub:shape}  {sscsw} {ssc1t}
996 \Mn@AddToConfig{sub:encoding/shape}{TS1/sw}{it}

```

Code for the last argument of \DeclareFontShape

```

997 \Mn@AddToConfig{code:shape}{sw}{
998     \skewchar\font='337
999 }

```

Declaration of font families and shapes

```

1000 \newcommand*\Mn@DeclareFontShape[6] [] {%

```

Check if any substitutions are specified.

```

1001     \edef\@tempa{%
1002         \Mn@UseConfig{sub:series}{#4}%
1003         \Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}%
1004         \Mn@UseConfig{sub:shape}{#5}%
1005     }%
1006     \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.

```

1007     \@temptokena=%
1008     \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
1009         \Mn@UseConfig{opticals}      {\Mn@option@opticals}%
1010         \Mn@UseConfig{fontset/weight}{\Mn@option@fontset/#4}%
1011         \Mn@UseConfig{weight}        {#4}%
1012         \Mn@UseConfig{encoding/shape}{#2/#5}%
1013         \Mn@UseConfig{shape}         {#5}%
1014     }%
1015     \edef\@tempa{\the\@temptokena{\Mn@TheConfig{code:shape}{#5}}}%
1016     \@tempa
1017     \else

```

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

```

1018     \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%

```



```

1019 <->ssub*#3-#6%
1020 /\Mn@UseConfigOrDefault{sub:series}{#4}{#4}%
1021 /\Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
1022 \Mn@UseConfigOrDefault{sub:shape}{#5}{#5}}%
1023 }{}%
1024 \fi
1025 }
1026 \otf@makeglobal{Mn@DeclareFontShape}
1027 \otf@makeglobal{\string\Mn@DeclareFontShape}

#2 contains the encoding, #3 the family, and #1 a list of figure versions (or Extra).
1028 \newcommand*\Mn@DeclareLargeFontFamily[3][LF,OsF,TLF,TOf]{%
1029 \Mn@DeclareFontFamily{#1}{#2}{#3}
1030 {m, sb, b, bx, eb} {n, it, sc, ssc, scit, sscit, sw, scsl, scsw, sscsl, sscsw, sl}%
1031 }
1032 \newcommand*\Mn@DeclareSmallFontFamily[3][LF,OsF,TLF,TOf]{%
1033 \Mn@DeclareFontFamily{#1}{#2}{#3}
1034 {m, sb, b, bx, eb} {n, it, sl}%
1035 }
1036 \newcommand*\Mn@DeclareMathFontFamily[3][TOf]{%
1037 \Mn@DeclareFontFamily[\skewchar\font=255]{#1}{#2}{#3}
1038 {m, sb, b, bx, eb} {n, it}%
1039 }

```

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

```

1040 \otf@makeglobal{Mn@DeclareLargeFontFamily}
1041 \otf@makeglobal{\string\Mn@DeclareLargeFontFamily}
1042 \otf@makeglobal{Mn@DeclareSmallFontFamily}
1043 \otf@makeglobal{\string\Mn@DeclareSmallFontFamily}
1044 \otf@makeglobal{Mn@DeclareMathFontFamily}
1045 \otf@makeglobal{\string\Mn@DeclareMathFontFamily}

1046 \newcommand*\Mn@DeclareFontFamily[6][ ]{%
1047 \@for\Mn@variant:=#2\do{%
1048 \DeclareFontFamily {#3}{#4-\Mn@variant}{#1}%
1049 }%
1050 \Mn@DeclareFontShapes{#3}{#4}
1051 {#5} {#6} {#2}%
1052 }
1053 \otf@makeglobal{Mn@DeclareFontFamily}
1054 \otf@makeglobal{\string\Mn@DeclareFontFamily}

1055 \newcommand*\Mn@DeclareFontShapes[5]{%
1056 \@for\Mn@series:=#3\do{%
1057 \@for\Mn@shape:=#4\do{%
1058 \@for\Mn@variant:=#5\do{%
1059 \Mn@DeclareFontShape{#1}{#2}{\Mn@series}{\Mn@shape}{\Mn@variant}%
1060 }%
1061 }%
1062 }%
1063 }

```

```

1064 \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).
1065 \newdimen\Mn@fontdimen
1066 \newcommand*\Mn@adjust@fontdimen[2]{%
1067 \Mn@fontdimen=\fontdimen#1\font
1068 #2%
1069 \fontdimen#1\font=\Mn@fontdimen
1070 }
1071 \otf@makeglobal {Mn@adjust@fontdimen}
1072 \ifx\nodocument\relax
1073 \endgroup
1074 \fi

1075 (*debug)
1076 \newcommand\old@DeclareFontFamily{}
1077 \let\old@DeclareFontFamily\DeclareFontFamily
1078 \renewcommand\DeclareFontFamily[3]{
1079 \begingroup\escapechar'\%
1080 \edef\@tempa{\noexpand\DeclareFontFamily{#1}{#2}}%
1081 \@temptokena\expandafter{\@tempa{#3}}%
1082 \message{\the\@temptokena}%
1083 \endgroup
1084 \old@DeclareFontFamily{#1}{#2}{#3}%
1085 }
1086 \newcommand\old@DeclareFontShape{}
1087 \let\old@DeclareFontShape\DeclareFontShape
1088 \renewcommand\DeclareFontShape[6]{
1089 \begingroup\escapechar'\%
1090 \edef\@tempa{\noexpand\DeclareFontShape{#1}{#2}{#3}{#4}{#5}}%
1091 \@temptokena\expandafter{\@tempa{#6}}%
1092 \message{\the\@temptokena}%
1093 \endgroup
1094 \old@DeclareFontShape{#1}{#2}{#3}{#4}{#5}{#6}%
1095 }
1096 /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.

```

1097 \gdef\Mn@MicroType@Aliases{%
1098 \DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}%
1099 \DeclareMicrotypeAlias{MinionPro-OsF}{MinionPro}%
1100 \DeclareMicrotypeAlias{MinionPro-TLF}{MinionPro}%
1101 \DeclareMicrotypeAlias{MinionPro-TOsF}{MinionPro}%
1102 }
1103 \@ifundefined{Microtype@Hook}{%
1104 \global\let\Microtype@Hook\Mn@MicroType@Aliases

```

```

1105 }{%
1106 \g@addto@macro\Microtype@Hook{\Mn@Microtype@Aliases}%
1107 }%
1108 \ifundefined{DeclareMicroTypeAlias}{\Mn@MicroType@Aliases}%
1109 /fontdef)

```

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

```

1110 (*fd)
1111 \input{MinionPro-FontDef.sty}%
1112 \Uextra \Mn@DeclareSmallFontFamily[Extra]{U} {MinionPro}
1113 \LGR \Mn@DeclareSmallFontFamily {LGR}{MinionPro}
1114 \LGI \Mn@DeclareSmallFontFamily {LGI}{MinionPro}
1115 \OT1 \Mn@DeclareLargeFontFamily {OT1}{MinionPro}
1116 \T1 \Mn@DeclareLargeFontFamily {T1} {MinionPro}
1117 \LY1 \Mn@DeclareLargeFontFamily {LY1}{MinionPro}
1118 \T5 \Mn@DeclareLargeFontFamily {T5} {MinionPro}
1119 \T2A \Mn@DeclareSmallFontFamily {T2A}{MinionPro}
1120 \T2B \Mn@DeclareSmallFontFamily {T2B}{MinionPro}
1121 \T2C \Mn@DeclareSmallFontFamily {T2C}{MinionPro}
1122 \TS1 \Mn@DeclareLargeFontFamily {TS1}{MinionPro}
1123 \X2 \Mn@DeclareSmallFontFamily {X2} {MinionPro}
1124 \OT2 \Mn@DeclareSmallFontFamily {OT2}{MinionPro}
1125 \OML & tosf \Mn@DeclareMathFontFamily {OML}{MinionPro}
1126 (*OML & (lf|osf|tlf))
1127 \@for\Mn@variant:=LF,TLF,OsF\do{%
1128 \DeclareFontFamily{OML}{MinionPro-\Mn@variant}{\skewchar\font=255}
1129 \@for\Mn@series:=m,sb,b,bx,eb\do{%
1130 \@for\Mn@shape:=n,it\do{%
1131 \DeclareFontShape{OML}{MinionPro-\Mn@variant}{\Mn@series}{\Mn@shape}%
1132 { <-> ssub*MinionPro-TOSF/\Mn@series/\Mn@shape }{}}
1133 }%
1134 }%
1135 }%
1136 /OML & (lf|osf|tlf))
1137 /fd)

```