MinionPro Support for LaTeX

Achim Blumensath Andreas Bühmann Sebastian Schubert
Michael Zedler

v2.3a – 2015/01/10

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 . . . 14
10.4 Greek letters . . . 14
10.5 pdfTeX to-unicode support . . . 16
10.6 Superior and inferior figures . . . 18
10.7 Additional symbols . . . 21
10.8 Integral symbols . . . 22
10.9 Open G and Round V support . . . 23
10.10 Logos . . . 23
10.11 AMS . . . 23

11 Support for character protrusion 24

12 Font definition files 29

1 Overview

The MinionPro package provides support for the MinionPro font family from Adobe. You can use these fonts in a 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). Options can be passed to these packages by either putting the corresponding `\usepackage` command before `\usepackage{MinionPro}` or by including 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*.

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

nonormalsize	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 nonormalsize option to do so.

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

onlytext	only change the text fonts
onlymath	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:

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

Calligraphic fonts

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

mnsy*	use the calligraphic font from MnSymbol: $\mathcal{ABC}$
cmsy	take the calligraphic symbols from Computer Modern: $\mathcal{ABC}$
swash	use the swash capitals from MinionPro: $\mathcal{ABC}$
abx	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.

amsbb*	use the AMS blackboard font: $\mathbb{NZQRC}$
fourierbb	use the Fourier blackboard font: $\mathbb{NZQRC}$
lucidabb	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.

mixedgreek*	uppercase Greek is upright, lowercase Greek is italic
italicgreek	all Greek letters are italic
frenchmath	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

scale=<*factor*>	scale the font size by <*factor*>
minionint	take the integral symbols from MinionPro, not from MnSymbol: $\int$ instead of $\int$
openg	use g instead of g in math mode.
loosequotes	The quote signs of MinionPro are set rather tight. This can lead to undesirable spacing for apostrophes. The loosequotes 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 `\-`.
footnotefigures	use special figures for footnote marks, i.e., example^{6,9} instead of example^{6,9}. This option can only be used if the footnote marks consist *solely* of figures. Note that if you use one of the KOMA-Script classes, customization of the footnotes via `\deffootnote` before loading this package will be overwritten.

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	0123456789	0123456789
tabular	0123456789	0123456789

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

text, osf	text figures

lining, lf	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`.

`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:

Ϝ	`\digamma`	ϰ	`\varkappa`	ϐ	`\varbeta`
϶	`\backepsilon`	ɜ	`\varbackepsilon`	ħ	`\hbar`
ℏ	`\hslash`	ƛ	`\lambdabar`	ƛ	`\lambdaslash`
ȷ	`\jmath`	ð	`\eth`	𝕜	`\Bbbk`
0̸	`\slashedzero`	ɡ	`\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 "1⁄12 litres of red wine" and can be accessed via

`\smallfrac{`⟨*numerator*⟩`}{`⟨*denominator*⟩`}`	$\frac{1}{3}$ $\frac{5}{17}$
`\slantfrac{`⟨*numerator*⟩`}{`⟨*denominator*⟩`}`	1⁄3 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	❦	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 pdfTeX	LF/TOsF, non-standard ligatures, swashes
1.001, 2.000	Ghostscript, pre-1.40 pdfTeX	LF/OsF/TOsF, ligatures, swashes, small caps
1.00x	Distiller, dvipdfmx	LF/TOsF
1.00x	pdfTeX 1.40	ok
2.000	Distiller, dvipdfmx, pdfTeX 1.40	ok

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

```
\input glyphtounicode
\pdfgentounicode=1
```

See the pdfTeX 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-TOsF, 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-TOsF, MinionPro-TLF	m, b (sb, bx), eb	n, it (sl)
OML	MinionPro-TOsF	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:

[1] via substitution in TS1 encoding

- added package options onlytext and onlymath
- added package option loosequotes
- added package option openg
- added package options normalsize and nonormalsize
- 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`, `\lambdaslash`
- 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

Version 2.2:

- add scale option
- fix typo in microtype hook

Version 2.3:

- fix[2] footnotefigures option with KOMA classes

Version 2.3a:

- remove[3] microtype warning concerning `\j`

10 The main style file

10.1 Options

```
⟨*style⟩
\newif\if@Mn@Text@
\newif\if@Mn@Math@
\@Mn@Text@true
\@Mn@Math@true
\RequirePackage{kvoptions}
\SetupKeyvalOptions{
  family = Mn,
  prefix = Mn@
}
```

[2] based on http://tex.stackexchange.com/a/54954/11605

[3] based on http://tex.stackexchange.com/a/222471/11605

```
\DeclareVoidOption{onlytext}{\@Mn@Text@true\@Mn@Math@false}
\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.

```
\DeclareStringOption[1.]{scale}
\newcommand\Mn@minionint@opticals{-NoOpticals}
\newcommand\Mn@minionint@bold{-Bold}
\DeclareVoidOption{slides}{%
  \def\Mn@minionint@opticals{-NoOpticals}%
  \PassOptionsToPackage{slides}{MinionPro-FontDef}}
\DeclareVoidOption{noopticals}{%
  \def\Mn@minionint@opticals{-NoOpticals}%
  \PassOptionsToPackage{noopticals}{MinionPro-FontDef}}
\DeclareVoidOption{opticals}{%
  \def\Mn@minionint@opticals{}%
  \PassOptionsToPackage{opticals}{MinionPro-FontDef}}
\DeclareVoidOption{smallfamily}{%
  \def\Mn@minionint@bold{-Bold}%
  \PassOptionsToPackage{smallfamily}{MinionPro-FontDef}}
\DeclareVoidOption{medfamily}{%
  \def\Mn@minionint@bold{-Semibold}%
  \PassOptionsToPackage{medfamily}{MinionPro-FontDef}}
\DeclareVoidOption{fullfamily}{%
  \def\Mn@minionint@bold{-Semibold}%
  \PassOptionsToPackage{fullfamily}{MinionPro-FontDef}}
\DeclareVoidOption{normalsize}{%
  \PassOptionsToPackage{normalsize}{MinionPro-FontDef}}
\DeclareVoidOption{nonormalsize}{%
  \PassOptionsToPackage{nonormalsize}{MinionPro-FontDef}}
```

Figure style

```
\newcommand\Mn@Text@Fig{OsF}
\newcommand\Mn@Math@Fig{OsF}
\newcommand\Mn@Text@Family{MinionPro-\Mn@Text@Fig}
\newcommand\Mn@Math@Family{MinionPro-\Mn@Math@Fig}
\newcommand\Mn@Math@TFamily{MinionPro-T\Mn@Math@Fig}
\newcommand\Mn@Math@LetterShape{it}
```

```
\DeclareVoidOption{textosf}{\def\Mn@Text@Fig{OsF}}
\DeclareVoidOption{textlf}{\def\Mn@Text@Fig{LF}}
\DeclareVoidOption{mathosf}{\def\Mn@Math@Fig{OsF}}
\DeclareVoidOption{mathlf}{\def\Mn@Math@Fig{LF}}
\DeclareVoidOption{osf}{\setkeys{Mn}{textosf,mathosf}}
\DeclareVoidOption{lf}{\setkeys{Mn}{textlf,mathlf}}
\DeclareVoidOption{mathtabular}{\let\Mn@Math@Family\Mn@Math@TFamily}
```

Calligraphic fonts

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

```
\newcommand\Mn@load@cal{}
\newcommand\Mn@load@bb{}
\newcommand\Mn@load@frak{}
```

Most options are handled by MnSymbol.

```
\DeclareVoidOption{mnsy}{
  \PassOptionsToPackage{mnsy}{MnSymbol}
  \def\Mn@load@cal{
    \SetMathAlphabet\mathcal{boldtabular}{OMS}{MnSymbolS}{b}{n}
  }
}
\DeclareVoidOption{cmsy}{
  \PassOptionsToPackage{cmsy}{MnSymbol}
  \def\Mn@load@cal{
    \SetMathAlphabet\mathcal{boldtabular}{OMS}{cmsy}{b}{n}
  }
}
\DeclareVoidOption{abx}{
  \PassOptionsToPackage{abx}{MnSymbol}
%  \def\Mn@load@cal{
%    \SetMathAlphabet\mathcal{boldtabular}{OT1}{mathc}{b}{n}
%  }
}
\DeclareVoidOption{swash}{
  \def\Mn@load@cal{
    \DeclareMathAlphabet\mathcal         {T1}{\Mn@Math@Family} {m} {sw}
    \SetMathAlphabet\mathcal{bold}       {T1}{\Mn@Math@Family} {eb}{sw}
    \SetMathAlphabet\mathcal{tabular}    {T1}{\Mn@Math@TFamily}{m} {sw}
    \SetMathAlphabet\mathcal{boldtabular}{T1}{\Mn@Math@TFamily}{eb}{sw}}
}
```

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.

```
\newcommand\Mn@load@greek{\Mn@greek@Mixed}
\DeclareVoidOption{frenchmath}{%
  \def\Mn@load@greek{\Mn@greek@Upright}%
  \def\Mn@Math@LetterShape{n}}
\DeclareVoidOption{mixedgreek}{%
  \def\Mn@load@greek{\Mn@greek@Mixed}}
\DeclareVoidOption{italicgreek}{%
  \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}).

```
\newcommand\Mn@load@amsbb{
  \let\mathbb\@undefined
  \let\Bbbk\@undefined
  \DeclareMathAlphabet\mathbb{U}{msb}{m}{n}
  \newcommand\Bbbk{\mathbb{\mathchar"717C}}}
\newcommand\Mn@load@lucidabb{
  \let\mathbb\@undefined
  \let\Bbbk\@undefined
  \DeclareFontFamily{U}{hlcm}{}
  \DeclareFontShape{U}{hlcm}{m}{n}{ <->s*[0.92] hlcra }{}
  \DeclareMathAlphabet\mathbb{U}{hlcm}{m}{n}
  \newcommand\Bbbk{\mathbb{k}}}
\newcommand\Mn@load@fourierbb{
  \let\mathbb\@undefined
  \let\Bbbk\@undefined
  \DeclareFontFamily{U}{futm}{}
  \DeclareFontShape{U}{futm}{m}{n}{ <->s*[0.95] fourier-bb }{}
  \DeclareMathAlphabet\mathbb{U}{futm}{m}{n}
  \newcommand\Bbbk{\mathbb{k}}}
\DeclareVoidOption{amsbb}{\let\Mn@load@bb\Mn@load@amsbb}
\DeclareVoidOption{lucidabb}{\let\Mn@load@bb\Mn@load@lucidabb}
\DeclareVoidOption{fourierbb}{\let\Mn@load@bb\Mn@load@fourierbb}
```

Integrals

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

Miscellaneous options

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

```
\DeclareVoidOption{footnotefigures}{%
  \def\@makefnmark{%
    \begingroup
    \normalfont
    \fontfamily{MinionPro-Extra}\fontencoding{U}\selectfont
    \@thefnmark
    \endgroup}%
  \@ifundefined{KOMAClassName}{}{\deffootnote[1em]{1.5em}{1em}{%
  \fontfamily{MinionPro-Extra}\fontencoding{U}\selectfont\thefootnotemark}}}
%
\newcommand\Mn@Define@Open@g{}
\DeclareVoidOption{openg}{%
  \def\Mn@Define@Open@g{%
```

```
    \mathcode‘g="8000%
    \DeclareMathSymbol{\Mn@g}{\mathalpha}{letters}{‘g}%
    \begingroup
    \lccode‘\~=‘\g
    \lowercase{\gdef~{\ifnum\the\mathgroup=\m@ne \openg \else \Mn@g \fi}}%
    \endgroup
  }}
%
\newcommand\Mn@Define@Round@v{}
\DeclareVoidOption{roundv}{%
  \def\Mn@Define@Round@v{%
    \mathcode‘v="8000%
    \DeclareMathSymbol{\Mn@v}{\mathalpha}{letters}{‘v}%
    \begingroup
    \lccode‘\~=‘\v
    \lowercase{\gdef~{\ifnum\the\mathgroup=\m@ne \upsilon \else \Mn@v \fi}}%
    \endgroup
  }}
%
\newcommand\Mn@Quote@Spacing{}
\DeclareVoidOption{loosequotes}{%
  \def\Mn@Quote@Spacing{\Mn@Quote@Spacing@Loose}}
```

Defaults

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

10.2 Font declarations

```
\RequirePackage{MinionPro-FontDef}
\@ifpackageloaded{textcomp}{}{\RequirePackage{textcomp}}

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

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

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

```
\if@Mn@Text@
  \edef\rmdefault{\Mn@Text@Family}
  \let\ibycusdefault\Mn@Text@Family
```

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

```
  \def\Mn@Quote@Spacing@Loose{%
    \@ifpackageloaded{microtype}{}{\RequirePackage[kerning=true]{microtype}}
    \@ifundefined{SetExtraKerning}{}{
      \let\Mn@Set@Quote@Spacing\SetExtraKerning}
%      \SetExtraKerning
%        [ unit = 1em ]
%        { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2,LY1},
%          family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
%          shape    = n }
%        { \textquotedblleft = {30,30},  \textquotedblright = {30,30},
%          \textquoteleft    = {30,30},  \textquoteright    = {30,30} }}
  }
  \newcommand*\Mn@Set@Quote@Spacing[3][]{}
  \Mn@Quote@Spacing
  \Mn@Set@Quote@Spacing
    [ unit = 1em ]
    { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2,LY1},
      family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
      shape    = {n,it} }
    { \textquotedblleft = {30,30},  \textquotedblright = {30,30},
      \textquoteleft    = {30,30},  \textquoteright    = {30,30} }
\fi
```

Math fonts

Redefine the standard math versions normal and bold.

```
\if@Mn@Math@
  \DeclareSymbolFont{operators}  {T1} {\Mn@Math@Family}{m} {n}
  \DeclareSymbolFont{letters}    {OML}{MinionPro-TOsF} {m} {\Mn@Math@LetterShape}
  \SetSymbolFont{operators}{bold}{T1} {\Mn@Math@Family}{eb}{n}
  \SetSymbolFont{letters}  {bold}{OML}{MinionPro-TOsF} {eb}{\Mn@Math@LetterShape}
  \DeclareMathAlphabet\mathbf    {T1} {\Mn@Math@Family}{eb}{n}
  \DeclareMathAlphabet\mathit    {T1} {\Mn@Math@Family}{m} {it}
  \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).

```
  \DeclareMathVersion{tabular}
  \SetSymbolFont{operators}{tabular}    {T1} {\Mn@Math@TFamily}{m}{n}
  \SetSymbolFont{letters}  {tabular}    {OML}{MinionPro-TOsF}  {m}{\Mn@Math@LetterShape}
  \SetMathAlphabet\mathit  {tabular}    {T1} {\Mn@Math@TFamily}{m}{it}

  \DeclareMathVersion{boldtabular}
  \SetSymbolFont{operators}{boldtabular}{T1} {\Mn@Math@TFamily}{eb}{n}
  \SetSymbolFont{letters}  {boldtabular}{OML}{MinionPro-TOsF}  {eb}{\Mn@Math@LetterShape}
  \SetMathAlphabet\mathit  {boldtabular}{T1} {\Mn@Math@TFamily}{eb}{it}
```

```
  \DeclareMathAccent{\grave}   {\mathalpha}{operators}{0}
  \DeclareMathAccent{\acute}   {\mathalpha}{operators}{1}
  \DeclareMathAccent{\hat}     {\mathalpha}{operators}{2}
  \DeclareMathAccent{\tilde}   {\mathalpha}{operators}{3}
  \DeclareMathAccent{\ddot}    {\mathalpha}{operators}{4}
  \DeclareMathAccent{\mathring}{\mathalpha}{operators}{6}
  \DeclareMathAccent{\check}   {\mathalpha}{operators}{7}
  \DeclareMathAccent{\breve}   {\mathalpha}{operators}{8}
  \DeclareMathAccent{\bar}     {\mathalpha}{operators}{9}
  \DeclareMathAccent{\dot}     {\mathalpha}{operators}{10}
```

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

```
  \Mn@load@bb
  \Mn@load@frak
  \Mn@load@cal
\fi
```

10.3 Font selection

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

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

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

```
\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.

```
\if@Mn@Math@
  \newcommand\Mn@greek@list@upper{}
  \newcommand\Mn@greek@list@lower{}
  \let\Mn@greek@list@upper\@gobble
  \let\Mn@greek@list@lower\@gobble
```

This macro holds one of the two list names.

```
  \newcommand\Mn@greek@list{}
  \newcommand*\Mn@greek@letter[3]{%
    \expandafter\DeclareMathSymbol
    \expandafter{\csname it#1\endcsname}{\mathord}{letters}{#2}%
    \expandafter\DeclareMathSymbol
    \expandafter{\csname up#1\endcsname}{\mathord}{letters}{#3}%
    \edef\@tempa{`\@car#1\@nil}%
    \edef\Mn@greek@list{\expandafter\noexpand\csname
      Mn@greek@list@\ifnum\uccode\@tempa=\@tempa upper\else lower\fi\endcsname}%
    \expandafter\edef\Mn@greek@list{\Mn@greek@list,#1}%
  }
```

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

```
  \Mn@greek@letter{Gamma}         {'000}{'200}
```

```
\Mn@greek@letter{Delta}         {'001}{'201}
\Mn@greek@letter{Theta}         {'002}{'202}
\Mn@greek@letter{Lambda}        {'003}{'203}
\Mn@greek@letter{Xi}            {'004}{'204}
\Mn@greek@letter{Pi}            {'005}{'205}
\Mn@greek@letter{Sigma}         {'006}{'206}
\Mn@greek@letter{Upsilon}       {'007}{'207}
\Mn@greek@letter{Phi}           {'010}{'210}
\Mn@greek@letter{Psi}           {'011}{'211}
\Mn@greek@letter{Omega}         {'012}{'212}
\Mn@greek@letter{alpha}         {'013}{'213}
\Mn@greek@letter{beta}          {'014}{'214}
\Mn@greek@letter{gamma}         {'015}{'215}
\Mn@greek@letter{delta}         {'016}{'216}
\Mn@greek@letter{epsilon}       {'017}{'217}
\Mn@greek@letter{zeta}          {'020}{'220}
\Mn@greek@letter{eta}           {'021}{'221}
\Mn@greek@letter{theta}         {'022}{'222}
\Mn@greek@letter{iota}          {'023}{'223}
\Mn@greek@letter{kappa}         {'024}{'224}
\Mn@greek@letter{lambda}        {'025}{'225}
\Mn@greek@letter{mu}            {'026}{'226}
\Mn@greek@letter{nu}            {'027}{'227}
\Mn@greek@letter{xi}            {'030}{'230}
\Mn@greek@letter{pi}            {'031}{'231}
\Mn@greek@letter{rho}           {'032}{'232}
\Mn@greek@letter{sigma}         {'033}{'233}
\Mn@greek@letter{tau}           {'034}{'234}
\Mn@greek@letter{upsilon}       {'035}{'235}
\Mn@greek@letter{phi}           {'036}{'236}
\Mn@greek@letter{chi}           {'037}{'237}
\Mn@greek@letter{psi}           {'040}{'240}
\Mn@greek@letter{omega}         {'041}{'241}
\Mn@greek@letter{varepsilon}    {'042}{'242}
\Mn@greek@letter{vartheta}      {'043}{'243}
\Mn@greek@letter{varpi}         {'044}{'244}
\Mn@greek@letter{varrho}        {'045}{'245}
\Mn@greek@letter{varsigma}      {'046}{'246}
\Mn@greek@letter{varphi}        {'047}{'247}
```

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

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

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

```
\newcommand*\Mn@greek@select[2]{%
  \expandafter\let\expandafter\Mn@greek@list\csname Mn@greek@list@#2\endcsname
```

```
    \@for\@tempa:=\Mn@greek@list\do{%
      \expandafter\let\csname\@tempa\expandafter\endcsname
      \csname#1\@tempa\endcsname
    }%
  }
  \newcommand*\Mn@greek@Upright{%
    \Mn@greek@select{up}{upper}%
    \Mn@greek@select{up}{lower}%
  }
  \newcommand*\Mn@greek@Italic{%
    \Mn@greek@select{it}{upper}%
    \Mn@greek@select{it}{lower}%
  }
  \newcommand*\Mn@greek@Mixed{%
    \Mn@greek@select{up}{upper}%
    \Mn@greek@select{it}{lower}%
  }
```

Finally initialise the Greek letters.

```
  \Mn@load@greek
\fi
```

10.5 pdfTEX to-unicode support

Old versions of MinionPro have non-standard glyph names.

```
\@ifundefined{pdfglyphtounicode}{}{
  \pdfglyphtounicode{uniEFD5}{03DD}% uni03DD
  \pdfglyphtounicode{uniEFED}{02D9}% dotaccent.cap
  \pdfglyphtounicode{uniEFEE}{02D8}% breve.cap
  \pdfglyphtounicode{uniEFF1}{02DB}% ogonek.cap
  \pdfglyphtounicode{uniEFF2}{00B8}% cedilla.cap
  \pdfglyphtounicode{uniEFF3}{02DA}% ring.cap
  \pdfglyphtounicode{uniEFF5}{02DC}% tilde.cap
  \pdfglyphtounicode{uniEFF7}{02C6}% circumflex.cap
  \pdfglyphtounicode{uniF628}{2030}% perthousand.oldstyle
  \pdfglyphtounicode{uniF62C}{0028}% parenleft.denominator
  \pdfglyphtounicode{uniF62D}{0029}% parenright.denominator
  \pdfglyphtounicode{uniF631}{0028}% parenleft.numerator
  \pdfglyphtounicode{uniF632}{0029}% parenright.numerator
  \pdfglyphtounicode{uniF638}{0030}% zero.slash
  \pdfglyphtounicode{uniF639}{0030}% zero.fitted
  \pdfglyphtounicode{uniF63A}{0032}% two.fitted
  \pdfglyphtounicode{uniF63B}{0033}% three.fitted
  \pdfglyphtounicode{uniF63C}{0034}% four.fitted
  \pdfglyphtounicode{uniF63D}{0035}% five.fitted
  \pdfglyphtounicode{uniF63E}{0036}% six.fitted
  \pdfglyphtounicode{uniF63F}{0037}% seven.fitted
  \pdfglyphtounicode{uniF640}{0038}% eight.fitted
  \pdfglyphtounicode{uniF641}{0039}% nine.fitted
  \pdfglyphtounicode{uniF642}{0025}% percent.oldstyle
```

```
\pdfglyphtounicode{uniF643}{0030}% zero.taboldstyle
\pdfglyphtounicode{uniF644}{0031}% one.taboldstyle
\pdfglyphtounicode{uniF645}{0032}% two.taboldstyle
\pdfglyphtounicode{uniF646}{0033}% three.taboldstyle
\pdfglyphtounicode{uniF647}{0034}% four.taboldstyle
\pdfglyphtounicode{uniF648}{0035}% five.taboldstyle
\pdfglyphtounicode{uniF649}{0036}% six.taboldstyle
\pdfglyphtounicode{uniF64A}{0037}% seven.taboldstyle
\pdfglyphtounicode{uniF64B}{0038}% eight.taboldstyle
\pdfglyphtounicode{uniF64C}{0039}% nine.taboldstyle
\pdfglyphtounicode{uniF64D}{20A1}% colonmonetary.taboldstyle
\pdfglyphtounicode{uniF64E}{20AC}% Euro.taboldstyle
\pdfglyphtounicode{uniF64F}{0192}% florin.taboldstyle
\pdfglyphtounicode{uniF650}{0023}% numbersign.taboldstyle
\pdfglyphtounicode{uniF651}{00A3}% sterling.taboldstyle
\pdfglyphtounicode{uniF652}{00A5}% yen.taboldstyle
\pdfglyphtounicode{uniF653}{0024}% dollar.taboldstyle
\pdfglyphtounicode{uniF654}{00A2}% cent.taboldstyle
\pdfglyphtounicode{uniF655}{0030}% zero.denominator
\pdfglyphtounicode{uniF656}{0031}% one.denominator
\pdfglyphtounicode{uniF657}{0032}% two.denominator
\pdfglyphtounicode{uniF658}{0033}% three.denominator
\pdfglyphtounicode{uniF659}{0034}% four.denominator
\pdfglyphtounicode{uniF65A}{0035}% five.denominator
\pdfglyphtounicode{uniF65B}{0036}% six.denominator
\pdfglyphtounicode{uniF65C}{0037}% seven.denominator
\pdfglyphtounicode{uniF65D}{0038}% eight.denominator
\pdfglyphtounicode{uniF65E}{0039}% nine.denominator
\pdfglyphtounicode{uniF65F}{002C}% comma.denominator
\pdfglyphtounicode{uniF660}{002E}% period.denominator
\pdfglyphtounicode{uniF661}{0030}% zero.numerator
\pdfglyphtounicode{uniF662}{0031}% one.numerator
\pdfglyphtounicode{uniF663}{0032}% two.numerator
\pdfglyphtounicode{uniF664}{0033}% three.numerator
\pdfglyphtounicode{uniF665}{0034}% four.numerator
\pdfglyphtounicode{uniF666}{0035}% five.numerator
\pdfglyphtounicode{uniF667}{0036}% six.numerator
\pdfglyphtounicode{uniF668}{0037}% seven.numerator
\pdfglyphtounicode{uniF669}{0038}% eight.numerator
\pdfglyphtounicode{uniF66A}{0039}% nine.numerator
\pdfglyphtounicode{uniF66B}{002C}% comma.numerator
\pdfglyphtounicode{uniF66C}{002E}% period.numerator
\pdfglyphtounicode{uniF66D}{0103}% abreve.sc
\pdfglyphtounicode{uniF66F}{0105}% aogonek.sc
\pdfglyphtounicode{uniF671}{0107}% cacute.sc
\pdfglyphtounicode{uniF672}{010D}% ccaron.sc
\pdfglyphtounicode{uniF675}{010F}% dcaron.sc
\pdfglyphtounicode{uniF676}{0111}% dcroat.sc
\pdfglyphtounicode{uniF678}{011B}% ecaron.sc
\pdfglyphtounicode{uniF67B}{014B}% eng.sc
```

```
  \pdfglyphtounicode{uniF67C}{0119}% eogonek.sc
  \pdfglyphtounicode{uniF67D}{011F}% gbreve.sc
  \pdfglyphtounicode{uniF684}{0133}% ij.sc
  \pdfglyphtounicode{uniF687}{0129}% itilde.sc
  \pdfglyphtounicode{uniF68A}{013A}% lacute.sc
  \pdfglyphtounicode{uniF68B}{013E}% lcaron.sc
  \pdfglyphtounicode{uniF68E}{0144}% nacute.sc
  \pdfglyphtounicode{uniF68F}{0148}% ncaron.sc
  \pdfglyphtounicode{uniF692}{0151}% ohungarumlaut.sc
  \pdfglyphtounicode{uniF695}{0155}% racute.sc
  \pdfglyphtounicode{uniF696}{0159}% rcaron.sc
  \pdfglyphtounicode{uniF698}{015B}% sacute.sc
  \pdfglyphtounicode{uniF699}{015F}% scedilla.sc
  \pdfglyphtounicode{uniF69D}{0165}% tcaron.sc
  \pdfglyphtounicode{uniF69E}{0163}% tcommaaccent.sc
  \pdfglyphtounicode{uniF6A0}{0171}% uhungarumlaut.sc
  \pdfglyphtounicode{uniF6A3}{016F}% uring.sc
  \pdfglyphtounicode{uniF6A4}{0169}% utilde.sc
  \pdfglyphtounicode{uniF6AA}{1EF3}% ygrave.sc
  \pdfglyphtounicode{uniF6AB}{017A}% zacute.sc
  \pdfglyphtounicode{uniF6AC}{017C}% zdotaccent.sc
  \pdfglyphtounicode{uniF6DC}{0031}% one.fitted
}
```

10.6 Superior and inferior figures

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

```
\def\@for@tok#1:=#2\do#3{%
  \expandafter\def\expandafter\@fortmp\expandafter{#2}%
  \ifx\@fortmp\@empty \else
    \expandafter\@forloop@tok#2\@nil\@nil\@@#1{#3}%
  \fi}
\def\@forloop@tok#1#2#3\@@#4#5{%
  \def#4{#1}%
  \ifx #4\@nnil \else
    #5%
    \def#4{#2}%
    \ifx #4\@nnil \else
      #5\@iforloop@tok #3\@@#4{#5}%
  \fi\fi}
\def\@iforloop@tok#1#2\@@#3#4{%
  \def#3{#1}%
  \ifx #3\@nnil
    \expandafter\@fornoop
  \else
    #4\relax\expandafter\@iforloop@tok
  \fi
  #2\@@#3{#4}}
%
```

```
\newcommand*\Mn@extra@font{%
  \fontencoding{U}\fontfamily{MinionPro-Extra}\selectfont}
\newcommand*\@numerator@fig[1]{{\Mn@extra@font\@@numerator@fig{#1}}}
\newcommand*\@denominator@fig[1]{{\Mn@extra@font\@@denominator@fig{#1}}}
\newcommand*\@superior@fig[1]{{\Mn@extra@font\@@superior@fig{#1}}}
\newcommand*\@inferior@fig[1]{{\Mn@extra@font\@@inferior@fig{#1}}}
\newcommand*\@@numerator@fig[1]{%
  \@for@tok\@nf@fig:=#1\do{%
    \ifcase\@nf@fig
       \char’00%
    \or\char’01%
    \or\char’02%
    \or\char’03%
    \or\char’04%
    \or\char’05%
    \or\char’06%
    \or\char’07%
    \or\char’10%
    \or\char’11%
    \else
      \@latex@error{invalid argument to \string\@@numerator@fig}%
    \fi
    }}
\newcommand*\@@denominator@fig[1]{%
  \@for@tok\@nf@fig:=#1\do{%
    \ifcase\@nf@fig
       \char’20%
    \or\char’21%
    \or\char’22%
    \or\char’23%
    \or\char’24%
    \or\char’25%
    \or\char’26%
    \or\char’27%
    \or\char’30%
    \or\char’31%
    \else
      \@latex@error{invalid argument to \string\@@denominator@fig}%
    \fi
    }}
\newcommand*\@@superior@fig[1]{%
  \@for@tok\@nf@fig:=#1\do{%
    \ifcase\@nf@fig
       \char’60%
    \or\char’61%
    \or\char’62%
    \or\char’63%
    \or\char’64%
    \or\char’65%
    \or\char’66%
    \or\char’67%
    \or\char’70%
    \or\char’71%
    \else
      \@latex@error{invalid argument to \string\@@superior@fig}%
    \fi
    }}
\newcommand*\@@inferior@fig[1]{%
  \@for@tok\@nf@fig:=#1\do{%
    \ifcase\@nf@fig
      \char’100%
    \or\char’101%
    \or\char’102%
    \or\char’103%
    \or\char’104%
    \or\char’105%
    \or\char’106%
    \or\char’107%
    \or\char’110%
    \or\char’111%
    \else
      \@latex@error{invalid argument to \string\@@inferior@fig}%
    \fi
    }}
```

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

```
\newcommand*\ensure@text[1]{%
    \ifmmode
      \Mn@Text@With@MathVersion{#1}%
    \else
      #1%
    \fi}
```

\smallfrac and \slantfrac assemble numerical fractions.

```
\newcommand*\@smallfrac[2]{%
    \leavevmode
    \setbox\@tempboxa
      \vbox{%
      \baselineskip\z@skip%
      \lineskip.25ex%
      \lineskiplimit-\maxdimen
      \ialign{\hfil##\hfil\crcr
              \vbox to 2.13ex{\vss\hbox{\@numerator@fig{#1}}\vskip.68ex}\crcr
              \leavevmode\leaders\hrule height 1.1ex depth -1.01ex\hfill\crcr
              \vtop to 1ex{\vbox{}\hbox{\@denominator@fig{#2}}\vss}\crcr
              \noalign{\vskip-1.47ex}}}%
    \dp\@tempboxa=0.49ex%
    \box\@tempboxa}
\newcommand*\@slantfrac[2]{%
  {\Mn@extra@font\@@numerator@fig{#1}\kern-0.05em/\kern-0.06em\@@denominator@fig{#2}}}
\DeclareRobustCommand*\smallfrac[2]{\ensure@text{\kern0.06em\@smallfrac{#1}{#2}\kern0.09em}}
```

```
\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.

```
\if@Mn@Math@
  \let\hbar\undefined
  \DeclareMathSymbol{\hbar}             {\mathord}{letters}{'265}
  \DeclareMathSymbol{\uphbar}           {\mathord}{letters}{'255}
  \DeclareMathSymbol{\partial}          {\mathord}{letters}{'100}
  \DeclareMathSymbol{\uppartial}        {\mathord}{letters}{'300}
  \DeclareMathSymbol{\ell}              {\mathord}{letters}{'140}
  \DeclareMathSymbol{\upell}            {\mathord}{letters}{'340}
  \DeclareMathSymbol{\slashedzero}      {\mathord}{letters}{'257}
  \DeclareMathSymbol{\upimath}          {\mathord}{letters}{'373}
  \DeclareMathSymbol{\upjmath}          {\mathord}{letters}{'374}
  \DeclareMathSymbol{\varsmallint}      {\mathord}{letters}{'376}
  \DeclareMathSymbol{\openg}            {\mathalpha}{letters}{'267}
  \DeclareRobustCommand\lambdabar       {\middlebar\lambda}
  \DeclareRobustCommand\lambdaslash     {\middleslash\lambda}
\fi
```

Archaic Greek letters not provided by MinionPro.

```
\if@Mn@Text@
  %\def\Qoppa{\reflectbox{P}}
  %\def\Sampi{\begingroup\fontfamily{cmr}\fontencoding{LGR}\selectfont\char23\endgroup}
  \let\Stigma\stigma

  % fix \r A
  \DeclareTextCompositeCommand{\r}{OT1}{A}
     {\leavevmode\setbox\z@\hbox{!}\dimen@\ht\z@\advance\dimen@-1ex%
     \ooalign{\hss\raise.67\dimen@\hbox{\char23}\hss\crcr A}}

  \DeclareEncodingSubset{TS1}{MinionPro-LF}  {1}%
  \DeclareEncodingSubset{TS1}{MinionPro-TLF} {1}%
  \DeclareEncodingSubset{TS1}{MinionPro-OsF} {1}%
  \DeclareEncodingSubset{TS1}{MinionPro-TOsF}{1}%
  \AtBeginDocument{
    \UndeclareTextCommand{\textvisiblespace}{T1}%
    \UndeclareTextCommand{\textcompwordmark}{T1}%
    \UndeclareTextCommand{\textsterling}{T1}%
    \UndeclareTextCommand{\j}{T1}%
    \UndeclareTextCommand{\j}{LY1}%
  }
\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.

```
\if@Mn@Math@
  \newcommand\Mn@Decl@Minion@Ints{%
```

Replace `MnSymbolF` by `MnSymbolFI`.

```
    \DeclareFontFamily{U}{MnSymbolFI}{}
    \DeclareFontShape{U}{MnSymbolFI}{m}{it}{
        <-6>  MnSymbolFI\Mn@minionint@opticals5
       <6-7>  MnSymbolFI\Mn@minionint@opticals6
       <7-8>  MnSymbolFI\Mn@minionint@opticals7
       <8-9>  MnSymbolFI\Mn@minionint@opticals8
       <9-10> MnSymbolFI\Mn@minionint@opticals9
      <10-12> MnSymbolFI\Mn@minionint@opticals10
      <12->   MnSymbolFI\Mn@minionint@opticals12
    }{}
    \DeclareFontShape{U}{MnSymbolFI}{b}{it}{
        <-6>  MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals5
       <6-7>  MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals6
       <7-8>  MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals7
       <8-9>  MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals8
       <9-10> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals9
      <10-12> MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals10
      <12->   MnSymbolFI\Mn@minionint@bold\Mn@minionint@opticals12
    }{}

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

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

```
    \let\varint\tint
    \let\variint\tiint
    \let\variiint\tiiint
    \let\variiiint\tiiiint
    \let\varidotsint\tidotsint
    \let\varlandupint\tlandupint
    \let\varlanddownint\tlanddownint
    \let\varstrokedint\tstrokedint
    \let\varoint\toint
    \let\varoiint\toiint
    \let\varrcirclerightint\trcirclerightint
    \let\varlcirclerightint\tlcirclerightint
    \let\varrcircleleftint\trcircleleftint
    \let\varlcircleleftint\tlcircleleftint
    \let\varsumint\tsumint
```

Replace the symbols with the new integrals.

```
    \DeclareMathSymbol\tint            \mathop{symbols}{112}
    \DeclareMathSymbol\tiint           \mathop{symbols}{114}
```

```
   \DeclareMathSymbol\tiiint          \mathop{symbols}{116}
   \DeclareMathSymbol\tiiiint         \mathop{symbols}{118}
   \DeclareMathSymbol\tidotsint       \mathop{symbols}{120}
   \DeclareMathSymbol\tlandupint      \mathop{symbols}{122}
   \DeclareMathSymbol\tlanddownint    \mathop{symbols}{124}
   \DeclareMathSymbol\tstrokedint     \mathop{symbols}{126}
   \DeclareMathSymbol\toint           \mathop{symbols}{128}
   \DeclareMathSymbol\toiint          \mathop{symbols}{130}
   \DeclareMathSymbol\trcirclerightint\mathop{symbols}{132}
   \DeclareMathSymbol\tlcirclerightint\mathop{symbols}{134}
   \DeclareMathSymbol\trcircleleftint \mathop{symbols}{136}
   \DeclareMathSymbol\tlcircleleftint \mathop{symbols}{138}
   \DeclareMathSymbol\tsumint         \mathop{symbols}{140}
   \let\intop\tint
   \let\ointop\toint
 }
 \Mn@load@integrals
\fi
```

10.9 Open G and Round V support

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

```
\if@Mn@Math@
 \Mn@Define@Open@g
 \Mn@Define@Round@v
\fi
```

10.10 Logos

Correct logos.

```
\if@Mn@Text@
 \def\TeX{T\kern-.1667em\lower.4ex\hbox{E}\kern-.125emX\@}
 \DeclareRobustCommand{\LaTeX}{L\kern-.32em%
         {\sbox\z@ T%
          \vbox to\ht\z@{\hbox{\check@mathfonts
                               \fontsize\sf@size\z@
                               \math@fontsfalse\selectfont
                               A}%
                         \vss}%
         }%
         \kern-.15em%
         \TeX}
\fi
```

10.11 AMS

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

```
\def\macc@set@skewchar#1{%
 \begingroup
 \ifnum\mathgroup=\m@ne \let\@tempa\@ne
 \else
  \ifnum\skewchar\textfont\mathgroup=\m@ne \let\@tempa\@ne
  \else \let\@tempa\mathgroup
  \fi
 \fi
 \count@=\skewchar\textfont\@tempa
 \ifnum\count@=\m@ne
  \endgroup
  \def\macc@skewchar{}
 \else
  \advance\count@"7100
  \edef\@tempa{\endgroup
   \mathchardef\noexpand\macc@skewchar=\number\count@\relax}%
  \@tempa
 \fi
 #1%
}
```

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

```
\if@Mn@Text@
 \normalfont
\fi
⟨/style⟩
```

11 Support for character protrusion

The microtype configuration. All four MinionPro families use the same file (cf. section 12). The inheritance tables are taken from microtype.cfg except `\j`.

```
⟨*mtcfg⟩
\DeclareCharacterInheritance
 { encoding = T1,
  family = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF} }
 { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
  a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
  C = {\'C,\c C,\v C},
  c = {\'c,\c c,\v c},
  D = {\v D,\DH},
  d = {\v d,\dj},
  E = {\`E,\'E,\^E,\"E,\k E,\v E},
  e = {\`e,\'e,\^e,\"e,\k e,\v e},
  f = {027}, % ff
  G = {\u G},
  g = {\u g},
  I = {\`I,\'I,\^I,\"I,\.I},
  i = {\`i,\'i,\^i,\"i,\i},
% j = {\j},
     L = {\L,\'L,\v L},
     l = {\l,\'l,\v l},
     N = {\'N,\~N,\v N},
     n = {\'n,\~n,\v n},
     O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
     o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
     R = {\'R,\v R},
     r = {\'r,\v r},
     S = {\'S,\c S,\v S,\SS},
     s = {\'s,\c s,\v s},
     T = {\c T,\v T},
     t = {\c t,\v t},
     U = {\`U,\'U,\^U,\"U,\H U,\r U},
     u = {\`u,\'u,\^u,\"u,\H u,\r u},
     Y = {\'Y,\"Y},
     y = {\'y,\"y},
     Z = {\'Z,\.Z,\v Z},
     z = {\'z,\.z,\v z}
   }
\SetProtrusion
  [ name     = MinionPro-OT1-Roman ]
  { encoding = OT1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = n }
  {
      A = {40,40},
      F = {  ,60},
      J = {90,  },
      K = {  ,50},
      L = {  ,60},
      T = {50,50},
      V = {40,40},
      W = {30,30},
      X = {50,50},
      Y = {50,50},
      k = {  ,60},
      r = {  ,80},
      t = {  ,100},
      v = {70,70},
      w = {40,40},
      x = {60,60},
      y = {70,70},
      ! = {70,180},
      ( = {60,30},    ) = {30,60},
      [ = {100,160},  ] = {160,100},
    {,} = {440,700},
      . = {660,700},
      : = {400,480},
      ; = {350,440},
      - = {700,700},
    \textendash       = {390,480},  \textemdash        = {220,270},
    \textquotedblleft = {380,250},  \textquotedblright = {250,380},
    \textquoteleft    = {670,450},  \textquoteright    = {450,670},
  }
\SetProtrusion
  [ name     = MinionPro-T1-Roman,
    load     = MinionPro-OT1-Roman ]
  { encoding = T1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = n }
  {
    023 = {  ,40}, % fft ligature
    032 = {  ,50}, % ft ligature
    191 = {30,30}, % Th ligature
    127 = {620,700}, % hyphen
    \AE = {40,  }, % AE
    \quotesinglbase = {670,670},  \quotedblbase   = {370,370},
    \guilsinglleft  = {500,360},  \guilsinglright = {360,500},
    \guillemotleft  = {320,230},  \guillemotright = {230,320},
  }
\SetProtrusion
  [ name     = MinionPro-OT1-Italic]
  { encoding = OT1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {it,sl,sw} }
  {
      A = {120,50},
      B = {90,-50},
      C = {50,-60},
      D = {70,-30},
      E = {90,-50},
      F = {100,-40},
      G = {50,-60},
      H = {70,-40},
      I = {150,-90},
      J = {250,-130},
      K = {80,-50},
      L = {90,60},
      M = {60,-40},
      N = {70,-40},
      O = {70,-30},
      P = {70,-110},
      Q = {40,-40},
      R = {80,-50},
      S = {70,-70},
      T = {130,  },
      U = {70,-40},
      V = {120,30},
      W = {90,20},
      X = {50,  },
      Y = {160,  },
      Z = {50,-50},
      d = {60,-60},
      f = {  ,-190},
    027 = {  ,-70}, % ff ligature
      g = {-70,-70},
      i = {  ,-110},
    025 = {  ,-60}, % dotlessi
    028 = {  ,-60}, % fi ligature
    030 = {  ,-30}, % ffi ligature
      j = {-90,-150},
      p = {-40,  },
      r = {  ,80},
      t = {  ,100},
      v = {90,  },
      w = {60,10},
      x = {90,  },
      ! = {190,40},
      ( = {90,  },    ) = {90,  },
      [ = {90,90},    ] = {120,60},
    {,} = {210,680},
      . = {640,680},
      : = {380,430},
      ; = {   ,430},
      - = {750,750},
    \textquoteleft    = {690,140},  \textquoteright    = {470,230},
    \textendash       = {400,500},  \textemdash        = {220,280},
    \textquotedblleft = {520,130},  \textquotedblright = {520,130},
  }
\SetProtrusion
  [ name     = MinionPro-T1-Italic,
    load     = MinionPro-OT1-Italic ]
  { encoding = T1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {it,sl,sw} }
  {
    023 = {  ,40}, % fft ligature
    032 = {  ,50}, % ft ligature
    191 = {80,30}, % Th ligature
    127 = {660,750}, % hyphen
    \AE = {90,-40}, % AE
    131 = {80,-30}, % Dcaron
    132 = {70,-40}, % Ecaron
    156 = {80,-60}, % IJ
    \OE = {50,-30}, % OE
    188 = {  ,-80}, % ij
    184 = {70,70}, % ydieresis
    253 = {70,70}, % yacute
    \quotesinglbase = {220,700},  \quotedblbase   = {130,400},
    \guilsinglleft  = {500,180},  \guilsinglright = {350,350},
    \guillemotleft  = {310,110},  \guillemotright = {230,230},
  }
```

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

```
\SetProtrusion
  [ name     = MinionPro-OT1-Smallcaps ]
  { encoding = OT1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {sc,ssc} }
  {}
\SetProtrusion
  [ name     = MinionPro-T1-Smallcaps,
    load     = MinionPro-OT1-Smallcaps ]
  { encoding = T1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {sc,ssc} }
  {}
\SetProtrusion
  [ name     = MinionPro-OT1-SmallcapsItalic ]
  { encoding = OT1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {scit,sscit} }
  {}
\SetProtrusion
  [ name     = MinionPro-T1-SmallcapsItalic,
    load     = MinionPro-OT1-SmallcapsItalic ]
  { encoding = T1,
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {scit,sscit} }
  {}
\SetProtrusion
  [ name     = MinionPro-other-Roman ]
  { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = n }
  {
      ! = {70,180},
      ( = {60,30},    ) = {30,60},
      [ = {100,160},  ] = {160,100},
    {,} = {440,700},
      . = {660,700},
      : = {400,480},
      ; = {350,440},
      - = {700,700},
    \textendash       = {390,480},  \textemdash        = {220,270},
    \textquotedblleft = {380,250},  \textquotedblright = {250,380},
    \textquoteleft    = {670,450},  \textquoteright    = {450,670},
  }
\SetProtrusion
  [ name     = MinionPro-other-Italic ]
  { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
    family   = {MinionPro-OsF,MinionPro-LF,MinionPro-TOsF,MinionPro-TLF},
    shape    = {it,sl,sw} }
  {
      ! = {190,40},
      ( = {90,  },    ) = {90,  },
      [ = {90,90},    ] = {120,60},
    {,} = {210,680},
      . = {640,680},
      : = {380,430},
      ; = {   ,430},
      - = {750,750},
    \textquoteleft    = {690,140},  \textquoteright    = {470,230},
    \textendash       = {400,500},  \textemdash        = {220,280},
    \textquotedblleft = {520,130},  \textquotedblright = {520,130},
  }
⟨/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.

```
⟨*fontdef⟩
\ifx\Mn@DeclareFontShape\@undefined\else\endinput\fi
```

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

```
\ifx\@nodocument\relax
  \input{otfontdef.sty}
\else
  \NeedsTeXFormat{LaTeX2e}
  \RequirePackage{otfontdef}
\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.

```
\ifx\@nodocument\relax
  \begingroup\escapechar`\\
\fi
```

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

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

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

```
\newif\ifMn@option@normalsize
\Mn@option@normalsizetrue
```

```
\ifx\@nodocument\relax\else
  \DeclareOption{slides}     {\let\Mn@option@opticals\CurrentOption}
  \DeclareOption{opticals}   {\let\Mn@option@opticals\CurrentOption}
  \DeclareOption{noopticals} {\let\Mn@option@opticals\CurrentOption}
  \DeclareOption{smallfamily}{\let\Mn@option@fontset\CurrentOption}
  \DeclareOption{medfamily}  {\let\Mn@option@fontset\CurrentOption}
  \DeclareOption{fullfamily} {\let\Mn@option@fontset\CurrentOption}
  \DeclareOption{normalsize}  {\Mn@option@normalsizetrue}
  \DeclareOption{nonormalsize}{\Mn@option@normalsizefalse}
  \ExecuteOptions{smallfamily,noopticals,normalsize}
  \ProcessOptions\relax
\fi
```

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

```
\ifMn@option@normalsize
  \begingroup
  \def\set@fontsize#1#2#3#4\@nil{%
    \@defaultunits\global\Mn@option@normalsize#2pt\relax\@nnil}%
  \normalsize\@nil
  \endgroup
\fi
```

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

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

Configuration database

```
\newcount\Mn@config@cnt
\Mn@config@cnt=0
\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.

```
\newcommand\Mn@AddToConfig{%
  \begingroup
  \nfss@catcodes
  \expandafter\endgroup
  \Mn@AddToConfig@
}
\newcommand\Mn@AddToConfig@[3]{%
  \advance\Mn@config@cnt\@ne
  \@namedef{\Mn@curr@config}{#3}%
  \otf@makeglobal{\Mn@curr@config}
⟨debug & show⟩\expandafter\show\csname\Mn@curr@config\endcsname
  \@for\Mn@tempa:=#2\do{%
    \@ifundefined{Mn@config@#1@\Mn@tempa}{%
      \@temptokena{}%
    }{%
      \@temptokena\expandafter\expandafter\expandafter
        {\csname Mn@config@#1@\Mn@tempa\endcsname}%
    }%
    \@expandtwoargs\@namedef{Mn@config@#1@\Mn@tempa}{%
      \the\@temptokena
      \expandafter\noexpand\csname\Mn@curr@config\endcsname
    }%
    \otf@makeglobal{Mn@config@#1@\Mn@tempa}% perhaps defer to only execute once
⟨debug & show⟩\expandafter\show\csname Mn@config@#1@\Mn@tempa\endcsname
  }%
}
```

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=l1]<->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.

```
\newcommand*\Mn@UseConfig[2]{%
  \Mn@UseConfigOrDefault{#1}{#2}{}%
}
\newcommand*\Mn@UseConfigOrDefault[3]{%
  \@ifundefined{Mn@config@#1@#2}{#3}%
    {\@nameuse{Mn@config@#1@#2}}%
}
\newcommand*\Mn@TheConfig[2]{%
  \@ifundefined{Mn@config@#1@#2}{}{%
```

```
    \expandafter\noexpand\csname Mn@config@#1@#2\endcsname
  }%
}
\otf@makeglobal{Mn@UseConfig}
\otf@makeglobal{Mn@UseConfigOrDefault}
\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.

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

Here comes the configuration.

```
\Mn@calc@bsize{\Mn@s@capt}{8.5}
\Mn@calc@bsize{\Mn@s@text}{13.1}
\Mn@calc@bsize{\Mn@s@subh}{20}
\Mn@AddToConfig{opticals}{opticals}{
            <-\Mn@s@capt>  otf* [optical=Capt]
  <\Mn@s@capt-\Mn@s@text>  otf* [optical=Text]
  <\Mn@s@text-\Mn@s@subh>  otf* [optical=Subh]
  <\Mn@s@subh->            otf* [optical=Disp]
}
\Mn@AddToConfig{opticals}{noopticals}{
      <->     otf* [optical=Text]
}
\Mn@AddToConfig{opticals}{slides}{
      <->     otf* [optical=Capt]
}
```

```
\ifdim\Mn@option@normalsize<10.1pt
  \Mn@calc@bsize{\Mn@s@semif}{6}
  \Mn@calc@bsize{\Mn@s@medif}{8.5}
\else
  \Mn@calc@bsize{\Mn@s@semif}{6}
  \Mn@calc@bsize{\Mn@s@medif}{10.1}
\fi
\Mn@AddToConfig{fontset/weight}{fullfamily/m}{
            < -\Mn@s@semif>  otf* [weight=Semibold]
  <\Mn@s@semif-\Mn@s@medif>  otf* [weight=Medium]
  <\Mn@s@medif->             otf* [weight=Regular]
}
\Mn@calc@bsize{\Mn@s@semim}{6}
\Mn@AddToConfig{fontset/weight}{medfamily/m}{
             <-\Mn@s@semim> otf* [weight=Semibold]
  <\Mn@s@semim->            otf* [weight=Regular]
}
\Mn@AddToConfig{fontset/weight}{smallfamily/m}{
      <->     otf* [weight=Regular]
}
```

```
%
\Mn@calc@bsize{\Mn@s@bold}{6}
\Mn@AddToConfig{fontset/weight}{fullfamily/b,medfamily/b}{
            <-\Mn@s@bold>  otf* [weight=Bold]
  <\Mn@s@bold->            otf* [weight=Semibold]
}
\Mn@AddToConfig{fontset/weight}{smallfamily/b}{
      <->     otf* [weight=Bold]
}
%
\Mn@AddToConfig{weight}{eb}{
      <->     otf* [weight=Bold]
}
\Mn@AddToConfig{shape}{ssc,sscit}{
      <->     otf* [spacing=l2]
}
\Mn@calc@bsize{\Mn@s@spac}{8}
\Mn@AddToConfig{shape}{n,it,sw,sc,scit}{
      <-\Mn@s@spac>    otf* [spacing=l1]
}
\Mn@AddToConfig{encoding/shape}{U/n,U/it}{
      <->     otf* [spacing=]
}
%
\Mn@AddToConfig{shape}{sc,ssc,scit,sscit}{
      <->     otf* [variant=sc]
}
\Mn@AddToConfig{shape}{sw}{
      <->     otf* [variant=swash]
}
\Mn@AddToConfig{shape}{it,scit,sscit,sw}{
      <->     otf* MinionPro-It
}
\Mn@AddToConfig{shape}{n,sc,ssc}{
      <->     otf* MinionPro
}
\Mn@AddToConfig{encoding/shape}{OML/it}{
      <->     otf* [figures=] MinionPro-Mixed
}
\Mn@AddToConfig{encoding/shape}{OML/n}{
      <->     otf* [figures=] MinionPro-French
}
\Mn@AddToConfig{scale}{scale}{
      <->     otf* [scale=\Mn@scale]
}
```

Substitutions

```
\Mn@AddToConfig{sub:series} {sb}    {b}
\Mn@AddToConfig{sub:series} {bx}    {b}
```

```
\Mn@AddToConfig{sub:shape}  {sl}    {it}
\Mn@AddToConfig{sub:shape}  {scsl}  {scit}
\Mn@AddToConfig{sub:shape}  {sscsl} {sscit}
\Mn@AddToConfig{sub:shape}  {scsw}  {scit}
\Mn@AddToConfig{sub:shape}  {sscsw} {sscit}
\Mn@AddToConfig{sub:encoding/shape}{TS1/sw}{it}
```

Code for the last argument of `\DeclareFontShape`

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

Declaration of font families and shapes

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

Check if any substitutions are specified.

```
  \edef\@tempa{%
    \Mn@UseConfig{sub:series}{#4}%
    \Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
      \Mn@UseConfig{sub:shape}{#5}}%
  }%
  \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.

```
    \@temptokena={%
      \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
        \Mn@UseConfig{opticals}      {\Mn@option@opticals}%
        \Mn@UseConfig{fontset/weight}{\Mn@option@fontset/#4}%
        \Mn@UseConfig{weight}        {#4}%
        \Mn@UseConfig{encoding/shape}{#2/#5}%
        \Mn@UseConfig{shape}         {#5}%
        \Mn@UseConfig{scale}         {scale}%
      }}%
    \edef\@tempa{\the\@temptokena{\Mn@TheConfig{code:shape}{#5}}}%
    \@tempa
  \else
```

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

```
    \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
      <->ssub*#3-#6%
      /\Mn@UseConfigOrDefault{sub:series}{#4}{#4}%
      /\Mn@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
        \Mn@UseConfigOrDefault{sub:shape}{#5}{#5}}%
    }{}%
  \fi
}
\otf@makeglobal{Mn@DeclareFontShape}
\otf@makeglobal{\string\Mn@DeclareFontShape}
```

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

```
\newcommand*\Mn@DeclareLargeFontFamily[3][LF,OsF,TLF,TOsF]{%
  \Mn@DeclareFontFamily{#1}{#2}{#3}
    {m,sb,b,bx,eb} {n,it,sc,ssc,scit,sscit,sw,scsl,scsw,sscsl,sscsw,sl}%
}
\newcommand*\Mn@DeclareSmallFontFamily[3][LF,OsF,TLF,TOsF]{%
  \Mn@DeclareFontFamily{#1}{#2}{#3}
    {m,sb,b,bx,eb} {n,it,sl}%
}
\newcommand*\Mn@DeclareMathFontFamily[3][TOsF]{%
  \Mn@DeclareFontFamily[\skewchar\font=255]{#1}{#2}{#3}
    {m,sb,b,bx,eb} {n,it}%
}
```

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

```
\otf@makeglobal{Mn@DeclareLargeFontFamily}
\otf@makeglobal{\string\Mn@DeclareLargeFontFamily}
\otf@makeglobal{Mn@DeclareSmallFontFamily}
\otf@makeglobal{\string\Mn@DeclareSmallFontFamily}
\otf@makeglobal{Mn@DeclareMathFontFamily}
\otf@makeglobal{\string\Mn@DeclareMathFontFamily}
```

```
\newcommand*\Mn@DeclareFontFamily[6][]{%
  \@for\Mn@variant:=#2\do{%
    \DeclareFontFamily {#3}{#4-\Mn@variant}{#1}%
  }%
  \Mn@DeclareFontShapes{#3}{#4}
    {#5} {#6} {#2}%
}
\otf@makeglobal{Mn@DeclareFontFamily}
\otf@makeglobal{\string\Mn@DeclareFontFamily}
```

```
\newcommand*\Mn@DeclareFontShapes[5]{%
  \@for\Mn@series:=#3\do{%
    \@for\Mn@shape:=#4\do{%
      \@for\Mn@variant:=#5\do{%
        \Mn@DeclareFontShape{#1}{#2}{\Mn@series}{\Mn@shape}{\Mn@variant}%
      }%
    }%
  }%
}
\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`).

```
\newdimen\Mn@fontdimen
\newcommand*\Mn@adjust@fontdimen[2]{%
  \Mn@fontdimen=\fontdimen#1\font
  #2%
  \fontdimen#1\font=\Mn@fontdimen
}
```

```
\otf@makeglobal{Mn@adjust@fontdimen}
\ifx\@nodocument\relax
  \endgroup
\fi
⟨*debug⟩
\newcommand\old@DeclareFontFamily{}
\let\old@DeclareFontFamily\DeclareFontFamily
\renewcommand\DeclareFontFamily[3]{
  \begingroup\escapechar`\\%
  \edef\@tempa{\noexpand\DeclareFontFamily{#1}{#2}}%
  \@temptokena\expandafter{\@tempa{#3}}%
  \message{\the\@temptokena}%
  \endgroup
  \old@DeclareFontFamily{#1}{#2}{#3}%
}
\newcommand\old@DeclareFontShape{}
\let\old@DeclareFontShape\DeclareFontShape
\renewcommand\DeclareFontShape[6]{
  \begingroup\escapechar`\\%
  \edef\@tempa{\noexpand\DeclareFontShape{#1}{#2}{#3}{#4}{#5}}%
  \@temptokena\expandafter{\@tempa{#6}}%
  \message{\the\@temptokena}%
  \endgroup
  \old@DeclareFontShape{#1}{#2}{#3}{#4}{#5}{#6}%
}
⟨/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.

```
\gdef\Mn@MicroType@Aliases{%
  \DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}%
  \DeclareMicrotypeAlias{MinionPro-OsF}{MinionPro}%
  \DeclareMicrotypeAlias{MinionPro-TLF}{MinionPro}%
  \DeclareMicrotypeAlias{MinionPro-TOsF}{MinionPro}%
}
\@ifundefined{Microtype@Hook}{%
  \global\let\Microtype@Hook\Mn@MicroType@Aliases
}{%
  \g@addto@macro\Microtype@Hook{\Mn@MicroType@Aliases}%
}%
\@ifundefined{DeclareMicroTypeAlias}{}{\Mn@MicroType@Aliases}%
⟨/fontdef⟩
```

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

```
⟨*fd⟩
\input{MinionPro-FontDef.sty}%
⟨Uextra⟩    \Mn@DeclareSmallFontFamily[Extra]{U}  {MinionPro}
⟨LGR⟩       \Mn@DeclareSmallFontFamily         {LGR}{MinionPro}
```

```
⟨LGI⟩          \Mn@DeclareSmallFontFamily       {LGI}{MinionPro}
⟨OT1⟩          \Mn@DeclareLargeFontFamily       {OT1}{MinionPro}
⟨T1⟩           \Mn@DeclareLargeFontFamily       {T1} {MinionPro}
⟨LY1⟩          \Mn@DeclareLargeFontFamily       {LY1}{MinionPro}
⟨T5⟩           \Mn@DeclareLargeFontFamily       {T5} {MinionPro}
⟨T2A⟩          \Mn@DeclareSmallFontFamily       {T2A}{MinionPro}
⟨T2B⟩          \Mn@DeclareSmallFontFamily       {T2B}{MinionPro}
⟨T2C⟩          \Mn@DeclareSmallFontFamily       {T2C}{MinionPro}
⟨TS1⟩          \Mn@DeclareLargeFontFamily       {TS1}{MinionPro}
⟨X2⟩           \Mn@DeclareSmallFontFamily       {X2} {MinionPro}
⟨OT2⟩          \Mn@DeclareSmallFontFamily       {OT2}{MinionPro}
⟨OML & tosf⟩   \Mn@DeclareMathFontFamily        {OML}{MinionPro}
⟨*OML & (lf | osf | tlf)⟩
  \@for\Mn@variant:=LF,TLF,OsF\do{%
    \DeclareFontFamily{OML}{MinionPro-\Mn@variant}{\skewchar\font=255}
    \@for\Mn@series:=m,sb,b,bx,eb\do{%
      \@for\Mn@shape:=n,it\do{%
        \DeclareFontShape{OML}{MinionPro-\Mn@variant}{\Mn@series}{\Mn@shape}%
          { <-> ssub*MinionPro-TOsF/\Mn@series/\Mn@shape }{}
      }%
    }%
  }%
⟨/OML & (lf | osf | tlf)⟩
⟨/fd⟩
```