MyriadPro Support for LaTeX

Sebastian Schubert

v0.1d – 2012/01/12

Contents

1 Overview **2**

2 Interference with other packages **2**

3 Options **3**

4 Additional mathversions sans and sansbold **4**

5 Figure selection and bold math symbols **5**

6 Additional font shapes and symbols **6**

7 Language support **6**

8 Searching for figures or for words containing ligatures in pdf documents **7**

9 nfss classification **7**

10 Version history **8**

11 The main style file **8**
11.1 Options 8
11.2 Font declarations 15
11.3 Font selection 18
11.4 Greek letters 18
11.5 pdfTeX to-unicode support 20
11.6 Superior and inferior figures 22
11.7 Additional symbols 24
11.8 Integral symbols 25
11.9 Logos 27
11.10AMS 27

12 Support for character protrusion **28**

1 Overview

The MyriadPro package provides support for the MyriadPro font family from Adobe. You can use these fonts in a LaTeX document by adding the command

```
\usepackage{MyriadPro}
```

to the preamble. This will change both the sans serif text font and the math font to MyriadPro. If you want to use MyriadPro as your main font, add

```
\renewcommand{\familydefault}{\sfdefault}
```

to your preamble. If you prefer another math font (such as eulervm), use the option onlytext as explained in Section 3. With the option sansmath, MyriadPro does not modify the main math fonts but defines a sans and sansbold mathversion, which use MyriadPro and MdSymbol. This allows the usage of a complete MyriadPro setup consisting of text and math to be used in only a part of the document. Load MyriadPro with sansmath after all other font packages (see Section 4)!

Acknowledgements

MyriadPro is heavily based on the MinionPro package by Achim Blumensath, Andreas Bühmann and Michael Zedler.

2 Interference with other packages

The MyriadPro package loads the following packages: textcomp, amsmath, fontaxes and mdsymbol. Do not load mdsymbol manually. If you want to pass options to the other packages, you can either put the corresponding `\usepackage` command before the `\usepackage{MyriadPro}` or you can include the options in the `\documentclass` command. The MyriadPro package is *not* compatible with amssymb and amsfonts. Please see also the corresponding section in the mdsymbol documentation.

The MyriadPro 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

The package also provides a way to only change the text fonts or only the math fonts. In addition, also additional font versions for sans serif math can be defined.

onlytext	only change the text fonts
onlymath	only change the math fonts
sansmath	provide mathversion *sans* and *sansbold* and change `\mathsf` to use MyriadPro. The other main math fonts are not modified. This can be used to only use MyriadPro's math in a part of the document (see Section 4).

Figure selection

MyriadPro offers four different figure versions. A detailed description is given in Section 5. 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.

cmsy*	take the calligraphic symbols from Computer Modern: $\mathcal{ABC}$
abx	use the calligraphic symbols provided by mathabx: $\mathcal{ABC}abc$ (This font contains also lowercase letters, but it is not quite finished.)
crswash[=*option*]	use the swash letters from CronosPro: $\mathcal{ABC}$. *option* can be either *noptsmall*, *optsmall*, *noptmed* or *optmed* using (no) optical weights, small or medium family configuration (see CronosPro documentation). First one is default.

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*>
loosequotes	The quote signs of MyriadPro 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.

4 Additional mathversions sans and sansbold

With the option sansmath, this package defines the additional mathversions sans and sansbold. They allow the usage of MyriadPro in math completely independent of the main math font. Also single input character symbols (e.g. `+`, `-`, `(`, `)`) adapt to the math version except when used with a delimiter size increasing command like `\big(`.[1] As a workaround, use the corresponding full command instead (`\big\lparen`) (see mdsymbol documentation).
Example: You want to use MyriadPro in table environments independently of the main

[1] Any help to solve this problem is highly welcome!

text and math fonts. Load MyriadPro with the sansmath option after all other font packages to define the additional math versions without modifying the main math font. Then use it in the following way:

```
\begin{table}
  \sffamily
  \mathversion{sans}
  ...
\end{table}
```

5 Figure selection and bold math symbols

MyriadPro 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

If you use the sansmath option, note that the `\figureversion` command does not check whether a sans mathversion is active. Switching to proportional or tabular figures always changes the mathversion to normal or tabular, respectively. If you want sans serif math, switch to mathversion sans or sanstabular after the call of `\figureversion`:

```
\mathversion{sans}        % sans serif math
...
\figureversion{tabular}   % switches to tabular figures in text
                          % and to mathversion tabular
\mathversion{sanstabular} % switch to sanstabular manually
...
```

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.

In addition to the `\mathsf` command, which produces bold symbols of Roman letters in math, MyriadPro offers the command `\boldsymbol`. It prints bold versions of Roman, Greek and other math symbols.
Example:
`\boldsymbol{A} \boldsymbol{+} \boldsymbol{\beta} = \boldsymbol{\mathcal{E}} \boldsymbol{\wedge} \boldsymbol{\mathrm{H}}` produces $\boldsymbol{A + \beta = \mathcal{E} \wedge \mathrm{H}}$.

6 Additional font shapes and symbols

The MyriadPro package provides all symbols from the mdsymbol package. Additionally, the following math symbols are available:

∅	`\slashedzero`	$\varkappa$	`\varkappa`	$\varbeta$	`\varbeta`
$\backepsilon$	`\backepsilon`	$\varbackepsilon$	`\varbackepsilon`	$\hbar$	`\hbar`
$\imath$	`\imath`	$\jmath$	`\jmath`	$\eth$	`\eth`
$\Bbbk$	`\Bbbk`				

Some of the alternative characters above resemble the normal character because MyriadPro offers no respective glyph. They are defined for compatibility reasons.

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

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

7 Language support

The following encodings are supported:

Latin	ot1, t1, ts1, ly1, t5
Cyrillic	t2a, t2b, t2c, x2, ot2
Greek	lgr (to be used with babel, including polutonikogreek), lgi (lbycus transliteration scheme)

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

`\usepackage[ibycus,⟨otherlanguages⟩]{babel}`

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

8 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
1.001, 2.000	Ghostscript, pre-1.40 pdfTeX	LF/OsF/TOsF, ligatures
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.

9 nfss classification

Parenthesised combinations are provided via substitutions.

encoding	family	series	shape
ot1, t1, ts1, ly1, t5	MyriadPro-OsF, MyriadPro-LF, MyriadPro-TOsF, MyriadPro-TLF	m, b (sb, bx), eb	n, it (sl)
lgr, lgi, t2a, t2b, t2c, x2, ot2	MyriadPro-OsF, MyriadPro-LF, MyriadPro-TOsF, MyriadPro-TLF	m, b (sb, bx), eb	n, it (sl)
oml	MyriadPro-TOsF	m, b (sb, bx), eb	n, it
u	MyriadPro-Extra	m, b (sb, bx), eb	n, it (sl)

10 Version history

Version 0.1: First version
Version 0.1a: Fixed onlytext option
Version 0.1b:
- Correction of mathfrak definition
- Correct mathversion sanstabular and sansboldtabular

Version 0.1c: Use down-case mdsymbol Version 0.1d: sansmath does not need onlytext

11 The main style file

11.1 Options

Set the default options. The given package options are taken into account after `\ProcessKeyvalOptions` below.

```
⟨*style⟩
\newif\if@My@Text@
\newif\if@My@Math@
\newif\if@My@Sans@Math@
\newif\if@My@Math@Symbols@
\@My@Text@true
\@My@Math@true
\@My@Sans@Math@false
\@My@Math@Symbols@false
\RequirePackage{kvoptions}
\SetupKeyvalOptions{
  family = My,
  prefix = My@
}
\DeclareVoidOption{onlytext}{\@My@Text@true\@My@Math@false}
\DeclareVoidOption{onlymath}{\@My@Text@false\@My@Math@true}
\DeclareVoidOption{sansmath}{\@My@Sans@Math@true\@My@Math@false}
```

Font sets

The package MyriadPro-FontDef adapts the font definitions to the requested font set (see section 13). So we simply pass on the relevant options including the font scale factor; only MyriadPro integrals are handled here in MyriadPro.

```
\DeclareStringOption[1.]{scale}
\newcommand\My@myriadint@opticals{-NoOpticals}
\newcommand\My@myriadint@bold{-Bold}
\newcommand\My@mdsym@regular{regular}
\newcommand\My@mdsym@bold{bold}
\DeclareVoidOption{noopticals}{%
  \def\My@myriadint@opticals{-NoOpticals}%
  \PassOptionsToPackage{noopticals}{MyriadPro-FontDef}}
\DeclareVoidOption{smallfamily}{%
```

```
  \def\My@myriadint@bold{-Bold}%
  \PassOptionsToPackage{smallfamily}{MyriadPro-FontDef}}
\DeclareVoidOption{medfamily}{%
  \def\My@myriadint@bold{-Semibold}%
  \def\My@mdsym@regular{autoregular}%
  \def\My@mdsym@bold{autosemibold}%
  \PassOptionsToPackage{medfamily}{MyriadPro-FontDef}}
%\DeclareVoidOption{fullfamily}{%
%  \def\My@myriadint@bold{-Semibold}%
%  \PassOptionsToPackage{fullfamily}{MyriadPro-FontDef}}
\DeclareVoidOption{normalsize}{%
  \PassOptionsToPackage{normalsize}{MyriadPro-FontDef}}
```

Figure style

```
\newcommand\My@Text@Fig{OsF}
\newcommand\My@Math@Fig{OsF}
\newcommand\My@Text@Family{MyriadPro-\My@Text@Fig}
\newcommand\My@Math@Family{MyriadPro-\My@Math@Fig}
\newcommand\My@Math@TFamily{MyriadPro-T\My@Math@Fig}
\newcommand\My@Math@LetterShape{it}
\newcommand\Cr@Math@Family{CronosPro-\My@Math@Fig}
\newcommand\Cr@Math@TFamily{CronosPro-T\My@Math@Fig}
\DeclareVoidOption{textosf}{\def\My@Text@Fig{OsF}}
\DeclareVoidOption{textlf}{\def\My@Text@Fig{LF}}
\DeclareVoidOption{mathosf}{\def\My@Math@Fig{OsF}}
\DeclareVoidOption{mathlf}{\def\My@Math@Fig{LF}}
\DeclareVoidOption{osf}{\setkeys{My}{textosf,mathosf}}
\DeclareVoidOption{lf}{\setkeys{My}{textlf,mathlf}}
\DeclareVoidOption{mathtabular}{\let\My@Math@Family\My@Math@TFamily}
```

Calligraphic fonts

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

```
\RequirePackage{fltpoint}
\fpDecimalSign{.}
\newcommand*{\My@calc@scale}[2]{\fpMul{#1}{#2}{\My@scale}}
\newcommand*{\My@calc@bsize}[2]{\fpDiv{#1}{#2}{\My@scale}}
\newcommand\My@load@cal{}
\newcommand\My@load@sans@cal{}
\newcommand\My@load@cal@both{}
\newcommand\My@load@bb{}
\newcommand\My@load@sans@bb{}
\newcommand\My@load@bb@both{}
\newcommand\My@load@frak{}
\newcommand\My@load@sans@frak{}
\newcommand\My@load@frak@both{}
\newcommand*\my@if@boldtabular@math[1]{%
  \@ifundefined{mv@boldtabular}{}{#1}%
}
```

Calligraphic fonts from Computer Modern:

```
\DeclareVoidOption{cmsy}{%
  \def\My@load@cal@both{%
    \My@calc@scale{\mdcmsy@scale}{0.99}
    \My@calc@bsize{\mdcmsy@scalea}{6.}
    \My@calc@bsize{\mdcmsy@scaleb}{7.}
    \My@calc@bsize{\mdcmsy@scalec}{8.}
    \My@calc@bsize{\mdcmsy@scaled}{9.}
    \My@calc@bsize{\mdcmsy@scalee}{10.}
    \DeclareFontFamily{OMS}{mdcmsy}{\skewchar\font48 }
    \DeclareFontShape{OMS}{mdcmsy}{m}{n}{%
      <              -\mdcmsy@scalea>s*[\mdcmsy@scale] cmsy5
      <\mdcmsy@scalea-\mdcmsy@scaleb>s*[\mdcmsy@scale] cmsy6
      <\mdcmsy@scaleb-\mdcmsy@scalec>s*[\mdcmsy@scale] cmsy7
      <\mdcmsy@scalec-\mdcmsy@scaled>s*[\mdcmsy@scale] cmsy8
      <\mdcmsy@scaled-\mdcmsy@scalee>s*[\mdcmsy@scale] cmsy9
      <\mdcmsy@scalee-              >s*[\mdcmsy@scale] cmsy10
    }{}
    \DeclareFontShape{OMS}{mdcmsy}{b}{n}{%
      <              -\mdcmsy@scaleb>s*[\mdcmsy@scale] cmbsy5
      <\mdcmsy@scaleb-\mdcmsy@scalee>s*[\mdcmsy@scale] cmbsy7
      <\mdcmsy@scalee-              >s*[\mdcmsy@scale] cmbsy10
    }{}
  }
  \def\My@load@cal{%
    \DeclareMathAlphabet{\mathcal}{OMS}{mdcmsy}{m}{n}%
    \SetMathAlphabet{\mathcal}{bold}{OMS}{mdcmsy}{b}{n}%
    \SetMathAlphabet{\mathcal}{boldtabular}{OMS}{mdcmsy}{b}{n}%
  }%
  \def\My@load@sans@cal{%
    \@ifundefined{mathcal}{%
      \DeclareMathAlphabet{\mathcal}{OMS}{mdcmsy}{m}{n}}
    \SetMathAlphabet{\mathcal}{sans}{OMS}{mdcmsy}{m}{n}%
    \SetMathAlphabet{\mathcal}{sansbold}{OMS}{mdcmsy}{b}{n}%
    \SetMathAlphabet{\mathcal}{sanstabular}{OMS}{mdcmsy}{m}{n}%
    \SetMathAlphabet{\mathcal}{sansboldtabular}{OMS}{mdcmsy}{b}{n}%
  }%
}
\DeclareVoidOption{abx}{%
  \def\My@load@cal@both{
    \My@calc@scale{\mdmathc@scale}{0.99}
    \DeclareFontFamily{OT1}{mdmathc}{}%
    \DeclareFontShape{OT1}{mdmathc}{m}{n}{ <->s*[\mdmathc@scale] mathc10 }{}%
  }
  \def\My@load@cal{%
    \DeclareMathAlphabet\mathcal{OT1}{mdmathc}{m}{n}%
  }%
  \def\My@load@sans@cal{%
    \@ifundefined{mathcal}{%
```

```
      \DeclareMathAlphabet{\mathcal}{OT1}{mdmathc}{m}{n}}%
    \SetMathAlphabet{\mathcal}{sans}{OT1}{mdmathc}{m}{n}%
    \SetMathAlphabet{\mathcal}{sansbold}{OT1}{mdmathc}{m}{n}%
  }%
}
\DeclareStringOption[false]{crswash}[noptsmall]
```

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}`).

```
\DeclareVoidOption{amsbb}{
  \def\My@load@bb@both{
    \My@calc@scale{\mdmsb@scale}{1.}
    \My@calc@bsize{\mdmsb@scalea}{6.}
    \My@calc@bsize{\mdmsb@scaleb}{7.}
    \My@calc@bsize{\mdmsb@scalec}{8.}
    \My@calc@bsize{\mdmsb@scaled}{9.}
    \My@calc@bsize{\mdmsb@scalee}{10.}
    \DeclareFontFamily{U}{mdmsb}{}
    \DeclareFontShape{U}{mdmsb}{m}{n}{%
      <              -\mdmsb@scalea>s*[\mdmsb@scale] msbm5%
      <\mdmsb@scalea-\mdmsb@scaleb>s*[\mdmsb@scale] msbm6%
      <\mdmsb@scaleb-\mdmsb@scalec>s*[\mdmsb@scale] msbm7%
      <\mdmsb@scalec-\mdmsb@scaled>s*[\mdmsb@scale] msbm8%
      <\mdmsb@scaled-\mdmsb@scalee>s*[\mdmsb@scale] msbm9%
      <\mdmsb@scalee-              >s*[\mdmsb@scale] msbm10%
    }{}
  }
  \def\My@load@bb{%
    \let\mathbb\@undefined%
    \let\Bbbk\@undefined%
    \DeclareMathAlphabet\mathbb{U}{mdmsb}{m}{n}%
    \newcommand\Bbbk{\mathbb{\mathchar"717C}}}
  \def\My@load@sans@bb{%
    \ifundef{\mathbb}{%
      \DeclareMathAlphabet\mathbb{U}{mdmsb}{m}{n}}{}%
    \SetMathAlphabet{\mathbb}{sans}{U}{mdmsb}{m}{n}%
    \SetMathAlphabet{\mathbb}{sansbold}{U}{mdmsb}{m}{n}%
    \SetMathAlphabet{\mathbb}{sanstabular}{U}{mdmsb}{m}{n}%
    \SetMathAlphabet{\mathbb}{sansboldtabular}{U}{mdmsb}{m}{n}%
    \mdsy@renewcommand{Bbbk}{\mathbb{\mathchar"717C}}}
}
\DeclareVoidOption{lucidabb}{
  \def\My@load@bb@both{
    \My@calc@scale{\mdhlcm@scale}{0.96}
    \DeclareFontFamily{U}{mdhlcm}{}
```

```
    \DeclareFontShape{U}{mdhlcm}{m}{n}{ <->s*[\mdhlcm@scale] hlcra }{}
  }
  \def\My@load@bb{
    \let\mathbb\@undefined
    \let\Bbbk\@undefined
    \DeclareMathAlphabet\mathbb{U}{mdhlcm}{m}{n}
    \newcommand\Bbbk{\mathbb{k}}}
  \def\My@load@sans@bb{
    \ifundef{\mathbb}{%
      \DeclareMathAlphabet\mathbb{U}{mdhlcm}{m}{n}}{}%
    \SetMathAlphabet{\mathbb}{sans}{U}{mdhlcm}{m}{n}%
    \SetMathAlphabet{\mathbb}{sansbold}{U}{mdhlcm}{m}{n}%
    \SetMathAlphabet{\mathbb}{sanstabular}{U}{mdhlcm}{m}{n}%
    \SetMathAlphabet{\mathbb}{sansboldtabular}{U}{mdhlcm}{m}{n}%
    \mdsy@renewcommand{Bbbk}{\mathbb{k}}}
}
\DeclareVoidOption{fourierbb}{
  \def\My@load@bb@both{
    \My@calc@scale{\mdfutm@scale}{0.99}
    \DeclareFontFamily{U}{mdfutm}{}
    \DeclareFontShape{U}{mdfutm}{m}{n}{ <->s*[\mdfutm@scale] four-
  ier-bb }{}
  }
  \def\My@load@bb{
    \let\mathbb\@undefined
    \let\Bbbk\@undefined
    \DeclareMathAlphabet\mathbb{U}{mdfutm}{m}{n}
    \newcommand\Bbbk{\mathbb{k}}}
  \def\My@load@sans@bb{
    \ifundef{\mathbb}{%
      \DeclareMathAlphabet\mathbb{U}{mdfutm}{m}{n}}{}%
    \SetMathAlphabet{\mathbb}{sans}{U}{mdfutm}{m}{n}%
    \SetMathAlphabet{\mathbb}{sansbold}{U}{mdfutm}{m}{n}%
    \SetMathAlphabet{\mathbb}{sanstabular}{U}{mdfutm}{m}{n}%
    \SetMathAlphabet{\mathbb}{sansboldtabular}{U}{mdfutm}{m}{n}%
    \mdsy@renewcommand{Bbbk}{\mathbb{k}}}
}
```

Fracture fonts

```
\def\My@load@frak@both{%
  \My@calc@scale{\mdeuf@scale}{1.}
  \My@calc@bsize{\mdeuf@scalea}{6.}
  \My@calc@bsize{\mdeuf@scaleb}{7.}
  \My@calc@bsize{\mdeuf@scalec}{8.}
  \My@calc@bsize{\mdeuf@scaled}{9.}
  \My@calc@bsize{\mdeuf@scalee}{10.}
  \DeclareFontFamily{U}{mdeuf}{}
  \DeclareFontShape{U}{mdeuf}{m}{n}{
    <              -\mdeuf@scaleb>s*[\mdeuf@scale] eufm5
    <\mdeuf@scaleb-\mdeuf@scalee>s*[\mdeuf@scale] eufm7
```

```
    <\mdeuf@scalee-             >s*[\mdeuf@scale] eufm10
  }{}
  \DeclareFontShape{U}{mdeuf}{b}{n}{
    <              -\mdeuf@scaleb>s*[\mdeuf@scale] eufb5
    <\mdeuf@scaleb-\mdeuf@scalee>s*[\mdeuf@scale] eufb7
    <\mdeuf@scalee-             >s*[\mdeuf@scale] eufb10
  }{}
}
\def\My@load@frak{%
  \DeclareMathAlphabet{\mathfrak}{U}{mdeuf}{m}{n}
  \SetMathAlphabet{\mathfrak}{bold}{U}{mdeuf}{b}{n}
  \SetMathAlphabet{\mathfrak}{boldtabular}{U}{mdeuf}{b}{n}
  \DeclareRobustCommand{\Re}{\mathfrak{R}}
  \DeclareRobustCommand{\Im}{\mathfrak{I}}
}
\def\My@load@sans@frak{%
  \ifundef{\mathfrak}{%
    \DeclareMathAlphabet{\mathfrak}{U}{mdeuf}{m}{n}%
    \SetMathAlphabet{\mathfrak}{bold}{U}{mdeuf}{b}{n}%
    \my@if@boldtabular@math{\SetMathAlphabet{\mathfrak}{boldtabular}{U}{mdeuf}{b}{n}
  }{}
  \@ifpackageloaded{eufrak}{%
    \SetMathAlphabet{\EuFrak}{sans}{U}{mdeuf}{m}{n}%
    \SetMathAlphabet{\EuFrak}{sansbold}{U}{mdeuf}{b}{n}%
    \SetMathAlphabet{\EuFrak}{sanstabular}{U}{mdeuf}{m}{n}%
    \SetMathAlphabet{\EuFrak}{sansboldtabular}{U}{mdeuf}{b}{n}%
  }{%
    \SetMathAlphabet{\mathfrak}{sans}{U}{mdeuf}{m}{n}%
    \SetMathAlphabet{\mathfrak}{sansbold}{U}{mdeuf}{b}{n}%
    \SetMathAlphabet{\mathfrak}{sanstabular}{U}{mdeuf}{m}{n}%
    \SetMathAlphabet{\mathfrak}{sansboldtabular}{U}{mdeuf}{b}{n}%
  }
  \mdsy@DeclareRobustCommand{Re}{\mathfrak{R}}
  \mdsy@DeclareRobustCommand{Im}{\mathfrak{I}}
}
```

Greek letters

`\My@greek@Upright`, `\My@greek@Mixed`, and `\My@greek@Italic` are defined below in section 11.4 before `\My@load@greek` is executed.

```
\newcommand\My@load@greek{\My@greek@Mixed}
\def\My@greek@upper{up}%
\def\My@greek@lower{it}%
\DeclareVoidOption{frenchmath}{%
  \def\My@greek@upper{up}%
  \def\My@greek@lower{up}%
  \def\My@Math@LetterShape{n}%
}
\DeclareVoidOption{mixedgreek}{%
```

```
  \def\My@greek@upper{up}%
  \def\My@greek@lower{it}%
}
\DeclareVoidOption{italicgreek}{%
  \def\My@greek@upper{it}%
  \def\My@greek@lower{it}%
}
```

Integrals

```
\newcommand\My@load@integrals{}
\DeclareVoidOption{myriadint}{\def\My@load@integrals{\My@Decl@Myriad@Ints}}
```

Miscellaneous options

Footnote figures, extra spacing for the apostrophe.

```
\DeclareVoidOption{footnotefigures}{%
  \def\@makefnmark{%
    \begingroup
    \normalfont
    \fontfamily{MyriadPro-Extra}\fontencoding{U}\selectfont
    \@thefnmark
    \endgroup}}
\newcommand\My@Quote@Spacing{}
\DeclareVoidOption{loosequotes}{%
  \def\My@Quote@Spacing{\My@Quote@Spacing@Loose}}
```

Defaults

```
\setkeys{My}{amsbb}
\setkeys{My}{cmsy}
\ProcessKeyvalOptions{My}\relax
\if@My@Math@
  \@My@Math@Symbols@true
\fi
\if@My@Sans@Math@
  \@My@Math@Symbols@true
\fi
\RequirePackage{ifthen}
\ifthenelse{\equal{\My@crswash}{false}}{}{%
  \def\My@load@cal@both{
    \My@calc@scale{\Cr@scale}{1.08}
    \ifthenelse{\equal{\My@crswash}{noptsmall}}{%
      \RequirePackage{CronosPro-FontDef}}{}
    \ifthenelse{\equal{\My@crswash}{optsmall}}{%
      \RequirePackage[opticals]{CronosPro-FontDef}}{}
    \ifthenelse{\equal{\My@crswash}{noptmed}}{%
      \RequirePackage[medfamily]{CronosPro-FontDef}}{}
    \ifthenelse{\equal{\My@crswash}{optmed}}{%
      \RequirePackage[opticals,medfamily]{CronosPro-FontDef}}{}}
```

```
  \def\My@load@cal{
    \DeclareMathAlphabet\mathcal          {T1}{\Cr@Math@Family} {m}{sw}
    \SetMathAlphabet\mathcal{bold}        {T1}{\Cr@Math@Family} {b}{sw}
    \SetMathAlphabet\mathcal{tabular}     {T1}{\Cr@Math@TFamily}{m}{sw}
    \SetMathAlphabet\mathcal{boldtabular}{T1}{\Cr@Math@TFamily}{b}{sw}}
  \def\My@load@sans@cal{
    \@ifundefined{mathcal}{%
      \DeclareMathAlphabet\mathcal            {T1}{\Cr@Math@Family}{m}{sw}}
    \SetMathAlphabet\mathcal{sans}            {T1}{\Cr@Math@Family}{m}{sw}
    \SetMathAlphabet\mathcal{sansbold}        {T1}{\Cr@Math@Family}{b}{sw}
    \SetMathAlphabet\mathcal{sanstabular}     {T1}{\Cr@Math@Family}{m}{sw}
    \SetMathAlphabet\mathcal{sansboldtabular}{T1}{\Cr@Math@Family}{b}{sw}}}
```

11.2 Font declarations

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

\if@My@Math@
  \DeclareMathVersion{tabular}
  \DeclareMathVersion{boldtabular}
  \RequirePackage[normalweight=\My@mdsym@regular,boldweight=\My@mdsym@bold,scale=\My@
\else
  \if@My@Sans@Math@
    \RequirePackage[normalweight=\My@mdsym@regular,boldweight=\My@mdsym@bold,scale=\M
  \fi
\fi
```

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

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

```
  \My@Set@Quote@Spacing
    [ unit = 1em ]
    { encoding = {OT1,T1,LGR,U,OT2,T2A,T2B,T2C,T5,X2,LY1},
      family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
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@My@Math@
  \DeclareSymbolFont{operators}  {T1} {\My@Math@Family}{m} {n}
  \DeclareSymbolFont{letters}    {OML}{MyriadPro-TOsF} {m} {\My@Math@LetterShape}
  \SetSymbolFont{operators}{bold}{T1} {\My@Math@Family}{b}{n}
  \SetSymbolFont{letters}  {bold}{OML}{MyriadPro-TOsF} {b}{\My@Math@LetterShape}
  \DeclareMathAlphabet\mathbf    {T1} {\My@Math@Family}{b}{n}
  \DeclareMathAlphabet\mathsf    {T1} {\My@Math@Family}{m} {n}
  \SetMathAlphabet\mathsf  {bold}{T1} {\My@Math@Family}{b}{n}
  \DeclareMathAlphabet\mathit    {T1} {\My@Math@Family}{m} {it}
  \SetMathAlphabet\mathit  {bold}{T1} {\My@Math@Family}{b}{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).

```
  \SetSymbolFont{operators}{tabular}    {T1} {\My@Math@TFamily}{m}{n}
  \SetSymbolFont{letters}  {tabular}    {OML}{MyriadPro-TOsF}  {m}{\My@Math@LetterSha
  \SetMathAlphabet\mathit  {tabular}    {T1} {\My@Math@TFamily}{m}{it}

  \SetSymbolFont{operators}{boldtabular}{T1} {\My@Math@TFamily}{b}{n}
  \SetSymbolFont{letters}  {boldtabular}{OML}{MyriadPro-TOsF}  {b}{\My@Math@LetterSha
  \SetMathAlphabet\mathit  {boldtabular}{T1} {\My@Math@TFamily}{b}{it}
```

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

```
  \My@load@bb@both
  \My@load@bb
  \My@load@frak@both
  \My@load@frak
  \My@load@cal@both
  \My@load@cal
\fi
```

Setup for sans serif math: set mathsf, create two new math versions for sans serif math and load correct swash letters.

```
\if@My@Sans@Math@

  \newcommand\IfSymbolFont[3]{\@ifundefined{sym#1}{#3}{#2}}

  \DeclareMathAlphabet\mathsf             {T1}{\My@Math@Family} {m}{n}
  \SetMathAlphabet\mathsf{bold}           {T1}{\My@Math@Family} {b}{n}
```

```
\SetMathAlphabet\mathsf{sansbold}       {T1}{\My@Math@Family} {b}{n}
\SetMathAlphabet\mathsf{sanstabular}    {T1}{\My@Math@TFamily}{m}{n}
\SetMathAlphabet\mathsf{sansboldtabular}{T1}{\My@Math@TFamily}{b}{n}

\SetMathAlphabet\mathit{sans}           {T1}{\My@Math@Family} {m}{it}
\SetMathAlphabet\mathit{sansbold}       {T1}{\My@Math@Family} {b}{it}
\SetMathAlphabet\mathit{sanstabular}    {T1}{\My@Math@TFamily}{m}{it}
\SetMathAlphabet\mathit{sansboldtabular}{T1}{\My@Math@TFamily}{b}{it}

\SetMathAlphabet\mathbf{sans}       {T1}{\My@Math@Family} {b}{n}
\SetMathAlphabet\mathbf{sanstabular}{T1}{\My@Math@TFamily}{b}{n}

\IfSymbolFont{operators}{%
  \SetSymbolFont{operators}{sans}{T1}{\My@Math@Family}{m}{n}
}{%
  \DeclareSymbolFont{operators}  {T1}{\My@Math@Family}{m}{n}
}
\SetSymbolFont{operators}{sansbold}       {T1}{\My@Math@Family} {b}{n}
\SetSymbolFont{operators}{sanstabular}    {T1}{\My@Math@TFamily}{m}{n}%
\SetSymbolFont{operators}{sansboldtabular}{T1}{\My@Math@TFamily}{b}{n}%

\IfSymbolFont{letters}{%
  \SetSymbolFont{letters}{sans}{OML}{MyriadPro-OsF}{r}{\My@Math@LetterShape}
}{%
  \DeclareSymbolFont{letters}  {OML}{MyriadPro-OsF}{r}{\My@Math@LetterShape}
}
\SetSymbolFont{letters}{sansbold}       {OML}{MyriadPro-OsF} {b}{\My@Math@LetterSha
\SetSymbolFont{letters}{sanstabular}    {OML}{MyriadPro-TOsF}{m}{\My@Math@LetterSha
\SetSymbolFont{letters}{sansboldtabular}{OML}{MyriadPro-TOsF}{b}{\My@Math@LetterSha

\My@load@cal@both
\My@load@sans@cal
\My@load@bb@both
\My@load@sans@bb
\My@load@frak@both
\My@load@sans@frak
```

Declare command to print a bold symbol of any math symbol. Code is taken from amsbsy to locally switch mathversion.

```
\mdsy@DeclareRobustCommandArg{boldsymbol}{1}{%
  \begingroup
  \let\@nomath\@gobble \mathversion{sansbold}%
  \math@atom{#1}{%
    \mathchoice%
    {\hbox{$\m@th\displaystyle#1$}}%
    {\hbox{$\m@th\textstyle#1$}}%
    {\hbox{$\m@th\scriptstyle#1$}}%
    {\hbox{$\m@th\scriptscriptstyle#1$}}}%
  \endgroup}
\fi
```

The accents are defined for math and/or sansmath.

```
\if@My@Math@Symbols@
  \mdsy@DeclareMathAccent{grave}   {\mathalpha}{operators}{0}
  \mdsy@DeclareMathAccent{acute}   {\mathalpha}{operators}{1}
  \mdsy@DeclareMathAccent{hat}     {\mathalpha}{operators}{2}
  \mdsy@DeclareMathAccent{tilde}   {\mathalpha}{operators}{3}
  \mdsy@DeclareMathAccent{ddot}    {\mathalpha}{operators}{4}
  \mdsy@DeclareMathAccent{mathring}{\mathalpha}{operators}{6}
  \mdsy@DeclareMathAccent{check}   {\mathalpha}{operators}{7}
  \mdsy@DeclareMathAccent{breve}   {\mathalpha}{operators}{8}
  \mdsy@DeclareMathAccent{bar}     {\mathalpha}{operators}{9}
  \mdsy@DeclareMathAccent{dot}     {\mathalpha}{operators}{10}
\fi
```

11.3 Font selection

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

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

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

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

11.4 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).

```
\if@My@Math@Symbols@
%    \begin{macrocode}
  \if@My@Sans@Math@
    \newcommand\My@greek@letter@[2]{
      \ifcsdef{#1}{%
        \csletcs{#1@old}{#1}%
      }{%
        \csletcs{#1@old}{#2#1}%
      }%
      \csletcs{sans#1}{#2#1}%
      \csundef{#1}%
      \csdef{#1}{\ifmathversionsans{\csname sans#1\endcsname}{\csname#1@old\endcsname
    }%
  \else
    \newcommand\My@greek@letter@[2]{%
      \csletcs{#1}{#2#1}
    }
  \fi
  \newcommand*\My@greek@letter[3]{%
    \mdsy@DeclareMathSymbol{it#1}{\mathord}{letters}{#2}%
```

```
    \mdsy@DeclareMathSymbol{up#1}{\mathord}{letters}{#3}%
    \edef\@tempa{`\@car#1\@nil}%
    \ifnum\uccode\@tempa=\@tempa%
      \My@greek@letter@{#1}{\My@greek@upper}%
    \else%
      \My@greek@letter@{#1}{\My@greek@lower}%
    \fi%
}
```

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

```
\My@greek@letter{Gamma}          {'000}{'200}
\My@greek@letter{Delta}          {'001}{'201}
\My@greek@letter{Theta}          {'002}{'202}
\My@greek@letter{Lambda}         {'003}{'203}
\My@greek@letter{Xi}             {'004}{'204}
\My@greek@letter{Pi}             {'005}{'205}
\My@greek@letter{Sigma}          {'006}{'206}
\My@greek@letter{Upsilon}        {'007}{'207}
\My@greek@letter{Phi}            {'010}{'210}
\My@greek@letter{Psi}            {'011}{'211}
\My@greek@letter{Omega}          {'012}{'212}
\My@greek@letter{alpha}          {'013}{'213}
\My@greek@letter{beta}           {'014}{'214}
\My@greek@letter{gamma}          {'015}{'215}
\My@greek@letter{delta}          {'016}{'216}
\My@greek@letter{epsilon}        {'017}{'217}
\My@greek@letter{zeta}           {'020}{'220}
\My@greek@letter{eta}            {'021}{'221}
\My@greek@letter{theta}          {'022}{'222}
\My@greek@letter{iota}           {'023}{'223}
\My@greek@letter{kappa}          {'024}{'224}
\My@greek@letter{lambda}         {'025}{'225}
\My@greek@letter{mu}             {'026}{'226}
\My@greek@letter{nu}             {'027}{'227}
\My@greek@letter{xi}             {'030}{'230}
\My@greek@letter{pi}             {'031}{'231}
\My@greek@letter{rho}            {'032}{'232}
\My@greek@letter{sigma}          {'033}{'233}
\My@greek@letter{tau}            {'034}{'234}
\My@greek@letter{upsilon}        {'035}{'235}
\My@greek@letter{phi}            {'036}{'236}
\My@greek@letter{chi}            {'037}{'237}
\My@greek@letter{psi}            {'040}{'240}
\My@greek@letter{omega}          {'041}{'241}
\My@greek@letter{varepsilon}     {'042}{'242}
\My@greek@letter{vartheta}       {'043}{'243}
\My@greek@letter{varpi}          {'044}{'244}
\My@greek@letter{varrho}         {'045}{'245}
\My@greek@letter{varsigma}       {'046}{'246}
\My@greek@letter{varphi}         {'047}{'247}
```

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

```
%%  \My@greek@letter{varbeta}       {'260}{'250}
  \My@greek@letter{varbeta}       {'014}{'214}
%%  \My@greek@letter{varkappa}      {'261}{'251}
  \My@greek@letter{varkappa}      {'024}{'224}
  \My@greek@letter{backepsilon}   {'262}{'252}
  \My@greek@letter{varbackepsilon}{'263}{'253}
  \My@greek@letter{digamma}       {'264}{'254}
  \My@greek@letter{eth}           {'266}{'256}
\fi
```

11.5 pdfTEX to-unicode support

Old versions of MyriadPro 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
}
```

11.6 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*\My@extra@font{%
  \fontencoding{U}\fontfamily{MyriadPro-Extra}\selectfont}
\newcommand*\My@numerator@fig[1]{{\My@extra@font\My@@numerator@fig{#1}}}
\newcommand*\My@denominator@fig[1]{{\My@extra@font\My@@denominator@fig{#1}}}
\newcommand*\My@superior@fig[1]{{\My@extra@font\My@@superior@fig{#1}}}
\newcommand*\My@inferior@fig[1]{{\My@extra@font\My@@inferior@fig{#1}}}
\newcommand*\My@@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\My@@numerator@fig}%
    \fi
    }}
\newcommand*\My@@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\My@@denominator@fig}%
    \fi
    }}
\newcommand*\My@@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\My@@superior@fig}%
    \fi
    }}
\newcommand*\My@@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\My@@inferior@fig}%
    \fi
    }}
```

`\Myensure@text` switches to text mode, if necessary.

```
\newcommand*\Myensure@text[1]{%
  \ifmmode
    \mdsy@text{#1}%
  \else
    #1%
  \fi}
```

`\smallfrac` and `\slantfrac` assemble numerical fractions.

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

11.7 Additional symbols

Some symbols missing from MdSymbol can be taken from MyriadPro.

```
\if@My@Math@Symbols@
  \mdsy@DeclareMathSymbol{hbar}              {\mathord}{letters}{'265}
```

```
  \mdsy@DeclareMathSymbol{uphbar}           {\mathord}{letters}{'255}
  \mdsy@DeclareMathSymbol{partial}          {\mathord}{letters}{'100}
  \mdsy@DeclareMathSymbol{uppartial}        {\mathord}{letters}{'300}
  \mdsy@DeclareMathSymbol{ell}              {\mathord}{letters}{'140}
  \mdsy@DeclareMathSymbol{upell}            {\mathord}{letters}{'340}
  \mdsy@DeclareMathSymbol{slashedzero}      {\mathord}{letters}{'257}
  \mdsy@DeclareMathSymbol{upimath}          {\mathord}{letters}{'373}
  \mdsy@DeclareMathSymbol{upjmath}          {\mathord}{letters}{'374}
  \mdsy@DeclareMathSymbol{varsmallint}      {\mathord}{letters}{'376}
\fi
```

Archaic Greek letters not provided by MyriadPro.

```
\if@My@Text@
  %\def\Qoppa{\reflectbox{P}}
  %\def\Sampi{\begingroup\fontfamily{cmr}\fontencoding{LGR}\selectfont\char23\endgrou
  \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}{MyriadPro-LF}  {1}%
  \DeclareEncodingSubset{TS1}{MyriadPro-TLF} {1}%
  \DeclareEncodingSubset{TS1}{MyriadPro-OsF} {1}%
  \DeclareEncodingSubset{TS1}{MyriadPro-TOsF}{1}%
  \AtBeginDocument{
    \UndeclareTextCommand{\textvisiblespace}{T1}%
    \UndeclareTextCommand{\textcompwordmark}{T1}%
    \UndeclareTextCommand{\textsterling}{T1}%
    \UndeclareTextCommand{\j}{T1}%
    \UndeclareTextCommand{\j}{LY1}%
  }
\fi
```

11.8 Integral symbols

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

```
\if@My@Math@
  \newcommand\My@Decl@Myriad@Ints{%
```

Replace `MdSymbolF` by `MySymbolFI`.

```
    \DeclareFontFamily{U}{MySymbolFI}{}
    \DeclareFontShape{U}{MySymbolFI}{m}{it}{
        <-6>  MySymbolFI\My@myriadint@opticals5
       <6-7>  MySymbolFI\My@myriadint@opticals6
       <7-8>  MySymbolFI\My@myriadint@opticals7
       <8-9>  MySymbolFI\My@myriadint@opticals8
       <9-10> MySymbolFI\My@myriadint@opticals9
```

```
      <10-12> MySymbolFI\My@myriadint@opticals10
      <12->   MySymbolFI\My@myriadint@opticals12
    }{}
    \DeclareFontShape{U}{MySymbolFI}{b}{it}{
        <-6>  MySymbolFI\My@myriadint@bold\My@myriadint@opticals5
       <6-7>  MySymbolFI\My@myriadint@bold\My@myriadint@opticals6
       <7-8>  MySymbolFI\My@myriadint@bold\My@myriadint@opticals7
       <8-9>  MySymbolFI\My@myriadint@bold\My@myriadint@opticals8
       <9-10> MySymbolFI\My@myriadint@bold\My@myriadint@opticals9
      <10-12> MySymbolFI\My@myriadint@bold\My@myriadint@opticals10
      <12->   MySymbolFI\My@myriadint@bold\My@myriadint@opticals12
    }{}
    \DeclareSymbolFont{symbols}  {U}{MySymbolFI}{m}{it}
    \SetSymbolFont{symbols}{bold}{U}{MySymbolFI}{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
```

```
  }
  \My@load@integrals
\fi
```

11.9 Logos

Correct logos.

```
\if@My@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
```

11.10 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@My@Text@
  \normalfont
```

```
\fi
⟨/style⟩
```

12 Support for character protrusion

The microtype configuration. All four MyriadPro families use the same file (cf. section 13).

```
⟨*mtcfg⟩
\SetProtrusion
  [ name     = MyriadPro-OT1-Roman ]
  { encoding = OT1,
    family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
  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     = MyriadPro-T1-Roman,
    load     = MyriadPro-OT1-Roman ]
  { encoding = T1,
```

```
    family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
  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     = MyriadPro-OT1-Italic]
  { encoding = OT1,
    family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
  TLF},
    shape    = {it,sl} }
  {
      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     = MyriadPro-T1-Italic,
    load     = MyriadPro-OT1-Italic ]
  { encoding = T1,
    family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
  TLF},
    shape    = {it,sl} }
  {
    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},
  }
\SetProtrusion
```

```
  [ name     = MyriadPro-other-Roman ]
  { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
    family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
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     = MyriadPro-other-Italic ]
  { encoding = {LGR,U,OT2,T2A,T2B,T2C,T5,X2},
    family   = {MyriadPro-OsF,MyriadPro-LF,MyriadPro-TOsF,MyriadPro-
TLF},
    shape    = {it,sl} }
  {
      ! = {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⟩
```

13 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 MyriadPro-FontDef.sty which is included by every fd file. Note that MyriadPro-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\My@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\My@option@opticals{noopticals}
\newcommand\My@option@fontset{smallfamily}
\newdimen\My@option@normalsize
\global\My@option@normalsize10pt
```

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

```
\newif\ifMy@option@normalsize
\My@option@normalsizetrue
```

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

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

```
\ifMy@option@normalsize
  \begingroup
  \def\set@fontsize#1#2#3#4\@nil{%
    \@defaultunits\global\My@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{My@option@opticals}
\otf@makeglobal{My@option@fontset}
\ifx\@nodocument\relax\else
  \PackageInfo{MyriadPro-FontDef}{%
    Configuration:\space\My@option@fontset,\space\My@option@opticals,\space
    normalsize=\the\My@option@normalsize}%
\fi
```

Configuration database

```
\newcount\My@config@cnt
\My@config@cnt=0
\newcommand\My@curr@config{My@config@\romannumeral\My@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\My@AddToConfig{%
  \begingroup
  \nfss@catcodes
  \expandafter\endgroup
  \My@AddToConfig@
}
\newcommand\My@AddToConfig@[3]{%
  \advance\My@config@cnt\@ne
  \@namedef{\My@curr@config}{#3}%
  \otf@makeglobal{\My@curr@config}
⟨debug & show⟩\expandafter\show\csname\My@curr@config\endcsname
  \@for\My@tempa:=#2\do{%
    \@ifundefined{My@config@#1@\My@tempa}{%
      \@temptokena{}%
    }{%
      \@temptokena\expandafter\expandafter\expandafter
        {\csname My@config@#1@\My@tempa\endcsname}%
    }%
    \@expandtwoargs\@namedef{My@config@#1@\My@tempa}{%
      \the\@temptokena
      \expandafter\noexpand\csname\My@curr@config\endcsname
    }%
    \otf@makeglobal{My@config@#1@\My@tempa}% perhaps defer to only ex-
  ecute once
⟨debug & show⟩\expandafter\show\csname My@config@#1@\My@tempa\endcsname
  }%
}
```

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*\My@UseConfig[2]{%
  \My@UseConfigOrDefault{#1}{#2}{}%
}
\newcommand*\My@UseConfigOrDefault[3]{%
  \@ifundefined{My@config@#1@#2}{#3}%
    {\@nameuse{My@config@#1@#2}}%
}
\newcommand*\My@TheConfig[2]{%
  \@ifundefined{My@config@#1@#2}{}{%
    \expandafter\noexpand\csname My@config@#1@#2\endcsname
  }%
}
\otf@makeglobal{My@UseConfig}
\otf@makeglobal{My@UseConfigOrDefault}
\otf@makeglobal{My@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{.}
\@ifundefined{My@calc@bsize}{%
\newcommand*{\My@calc@bsize}[2]{\fpDiv{#1}{#2}{\My@scale}}}
```

Here comes the configuration.

```
\My@calc@bsize{\My@s@capt}{8.5}
\My@calc@bsize{\My@s@text}{13.1}
\My@calc@bsize{\My@s@subh}{20}
\My@AddToConfig{opticals}{opticals}{
             <-\My@s@capt>  otf* [optical=Capt]
  <\My@s@capt-\My@s@text>  otf* [optical=Text]
  <\My@s@text-\My@s@subh>  otf* [optical=Subh]
  <\My@s@subh->            otf* [optical=Disp]
}
\My@AddToConfig{opticals}{noopticals}{
      <->     otf* [optical=Text]
}
\My@AddToConfig{opticals}{slides}{
      <->     otf* [optical=Capt]
}
\My@calc@bsize{\My@s@semim}{6}
\My@AddToConfig{fontset/weight}{medfamily/m}{
             <-\My@s@semim> otf* [weight=Semibold]
  <\My@s@semim->            otf* [weight=Regular]
}
\My@AddToConfig{fontset/weight}{smallfamily/m}{
      <->     otf* [weight=Regular]
}
%
\My@calc@bsize{\My@s@bold}{6}
```

```
\My@AddToConfig{fontset/weight}{fullfamily/b,medfamily/b}{
             <-\My@s@bold>  otf* [weight=Bold]
  <\My@s@bold->             otf* [weight=Semibold]
}
\My@AddToConfig{fontset/weight}{smallfamily/b}{
       <->     otf* [weight=Bold]
}
%
\My@AddToConfig{weight}{eb}{
       <->     otf* [weight=Bold]
}
\My@calc@bsize{\My@s@spac}{8}
\My@AddToConfig{shape}{n,it}{
       <-\My@s@spac>    otf* [spacing=l1]
}
\My@AddToConfig{encoding/shape}{U/n,U/it}{
       <->     otf* [spacing=]
}
\My@AddToConfig{shape}{it}{
       <->     otf* MyriadPro-It
}
\My@AddToConfig{shape}{n}{
       <->     otf* MyriadPro
}
\My@AddToConfig{encoding/shape}{OML/it}{
       <->     otf* [figures=] MyriadPro-Mixed
}
\My@AddToConfig{encoding/shape}{OML/n}{
       <->     otf* [figures=] MyriadPro-French
}
\My@AddToConfig{scale}{scale}{
       <->     otf* [scale=\My@scale]
}
```

Substitutions

```
\My@AddToConfig{sub:series} {sb}    {b}
\My@AddToConfig{sub:series} {bx}    {b}
\My@AddToConfig{sub:shape}  {sl}    {it}
```

Code for the last argument of `\DeclareFontShape`
Declaration of font families and shapes

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

Check if any substitutions are specified.

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

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

```
    \@temptokena={%
      \DeclareFontShape{#2}{#3-#6}{#4}{#5}{%
        \My@UseConfig{opticals}      {\My@option@opticals}%
        \My@UseConfig{fontset/weight}{\My@option@fontset/#4}%
        \My@UseConfig{weight}        {#4}%
        \My@UseConfig{encoding/shape}{#2/#5}%
        \My@UseConfig{shape}         {#5}%
        \My@UseConfig{scale}         {scale}%
      }}%
    \edef\@tempa{\the\@temptokena{\My@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%
      /\My@UseConfigOrDefault{sub:series}{#4}{#4}%
      /\My@UseConfigOrDefault{sub:encoding/shape}{#2/#5}{%
        \My@UseConfigOrDefault{sub:shape}{#5}{#5}}%
    }{}%
  \fi
}
\otf@makeglobal{My@DeclareFontShape}
\otf@makeglobal{\string\My@DeclareFontShape}
```

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

```
\newcommand*\My@DeclareLargeFontFamily[3][LF,OsF,TLF,TOsF]{%
  \My@DeclareFontFamily{#1}{#2}{#3}
    {m,sb,b,bx,eb} {n,it,sl}%
}
\newcommand*\My@DeclareSmallFontFamily[3][LF,OsF,TLF,TOsF]{%
  \My@DeclareFontFamily{#1}{#2}{#3}
    {m,sb,b,bx,eb} {n,it,sl}%
}
\newcommand*\My@DeclareMathFontFamily[3][TOsF]{%
  \My@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{My@DeclareLargeFontFamily}
\otf@makeglobal{\string\My@DeclareLargeFontFamily}
\otf@makeglobal{My@DeclareSmallFontFamily}
\otf@makeglobal{\string\My@DeclareSmallFontFamily}
\otf@makeglobal{My@DeclareMathFontFamily}
\otf@makeglobal{\string\My@DeclareMathFontFamily}
\newcommand*\My@DeclareFontFamily[6][]{%
```

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

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

```
\newdimen\My@fontdimen
\newcommand*\My@adjust@fontdimen[2]{%
  \My@fontdimen=\fontdimen#1\font
  #2%
  \fontdimen#1\font=\My@fontdimen
}
\otf@makeglobal{My@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 MyriadPro family variants into one microtype file: mt-MyriadPro.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\My@MicroType@Aliases{%
  \DeclareMicrotypeAlias{MyriadPro-LF}{MyriadPro}%
  \DeclareMicrotypeAlias{MyriadPro-OsF}{MyriadPro}%
  \DeclareMicrotypeAlias{MyriadPro-TLF}{MyriadPro}%
  \DeclareMicrotypeAlias{MyriadPro-TOsF}{MyriadPro}%
}
\@ifundefined{Microtype@Hook}{%
  \global\let\Microtype@Hook\My@MicroType@Aliases
}{%
  \g@addto@macro\Microtype@Hook{\My@MicroType@Aliases}%
}%
\@ifundefined{DeclareMicroTypeAlias}{}{\My@MicroType@Aliases}%
⟨/fontdef⟩
```

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

```
⟨*fd⟩
\input{MyriadPro-FontDef.sty}%
⟨Uextra⟩     \My@DeclareSmallFontFamily[Extra]{U}  {MyriadPro}
⟨LGR⟩         \My@DeclareSmallFontFamily       {LGR}{MyriadPro}
⟨LGI⟩        \My@DeclareSmallFontFamily       {LGI}{MyriadPro}
⟨OT1⟩         \My@DeclareLargeFontFamily       {OT1}{MyriadPro}
⟨T1⟩          \My@DeclareLargeFontFamily       {T1} {MyriadPro}
⟨LY1⟩        \My@DeclareLargeFontFamily       {LY1}{MyriadPro}
⟨T5⟩          \My@DeclareLargeFontFamily       {T5} {MyriadPro}
⟨T2A⟩         \My@DeclareSmallFontFamily       {T2A}{MyriadPro}
⟨T2B⟩         \My@DeclareSmallFontFamily       {T2B}{MyriadPro}
⟨T2C⟩         \My@DeclareSmallFontFamily       {T2C}{MyriadPro}
⟨TS1⟩        \My@DeclareLargeFontFamily       {TS1}{MyriadPro}
⟨X2⟩          \My@DeclareSmallFontFamily       {X2} {MyriadPro}
⟨OT2⟩         \My@DeclareSmallFontFamily       {OT2}{MyriadPro}
⟨OML & tosf⟩  \My@DeclareMathFontFamily        {OML}{MyriadPro}
⟨*OML & (lf ⋕ osf ⋕ tlf)⟩
  \@for\My@variant:=LF,TLF,OsF\do{%
    \DeclareFontFamily{OML}{MyriadPro-\My@variant}{\skewchar\font=255}
    \@for\My@series:=m,sb,b,bx,eb\do{%
      \@for\My@shape:=n,it\do{%
        \DeclareFontShape{OML}{MyriadPro-\My@variant}{\My@series}{\My@shape}%
          { <-> ssub*MyriadPro-TOsF/\My@series/\My@shape }{}
      }%
    }%
```

```
1311     }%
1312 ⟨/OML & (lf ⨌ osf ⨌ tlf)⟩
1313 ⟨/fd⟩
```