

# Axes, axes, axes

Andreas Bühmann

v0.2b – 2007/03/31

## Abstract

The package `fontaxes` simulates multiple independent font selection axes on top of certain single NFSS axes: *base family*, *figure style*, and *figure alignment* on top of *family*; *primary shape* and *secondary shape* on top of *shape*; and *math weight* and *math figure alignment* on top of *math version*.

## Contents

<b>1</b>	<b>Usage</b>	<b>2</b>
1.1	Shape . . . . .	2
1.2	Figure version . . . . .	2
1.3	Math version . . . . .	2
1.4	Additional commands . . . . .	3
<b>2</b>	<b>Naming conventions</b>	<b>4</b>
<b>3</b>	<b>Implementation</b>	<b>4</b>
3.1	High-level author commands (Level 1) . . . . .	4
3.1.1	Shape . . . . .	4
3.1.2	Figure version . . . . .	5
3.1.3	Math version . . . . .	6
3.2	Low-level author commands (Level 2) . . . . .	6
3.3	Internals (Layer 3) . . . . .	7
3.4	Encoding . . . . .	9
3.5	Decoding . . . . .	11
3.6	Compatibility . . . . .	12
3.7	Tools . . . . .	12
3.8	Tests . . . . .	14

# 1 Usage

## 1.1 Shape

fontaxes splits L<sup>A</sup>T<sub>E</sub>X's single shape axis into two ones: the primary shape axis (**n**, **it**, etc.) and the secondary shape axis (**ulc**, **sc**, etc.)<sup>1</sup>.

The customary commands **\upshape**, **\itshape**, and **\slshape** are redefined to access the primary axis only. For access to a swash shape the command **\swshape** is added.

The commands **\scshape** and **\sscshape** (spaced small caps) access the secondary axis. To return from any small-caps shape to upper and lower case a command **\ulcshape** is introduced.

All these commands update the shape axes using the low-level commands **\fontprimaryshape{<value>}** and **\fontsecondaryshape{<value>}**.

If you would like to change which values are used by the various commands **\<abbr>shape**, redefine the corresponding **\<abbr>default**. The additional **\swdefault**, **\sscdefault**, and **\ulcdefault** are provided with their default values **sw**, **ssc**, and **ulc**.

## 1.2 Figure version

Different figure versions are often implemented as additional families (e.g., MinionPro{-OsF,-LF,-T0sF,-TLF}<sup>2</sup>; or pplj, pplx). fontaxes splits off the axes *figure style* and *figure alignment*, which leaves the *base family* (MinionPro or ppl).

fontaxes knows two figure styles, **text** and **lining** (accessible via **\txfigures** and **\lnfigures**), and two modes of figure alignment, **tabular** and **proportional** (accessible via the switches **\tbfigures** and **\prfigures**).

Additionally, you can access both axes directly using the low-level commands **\fontfigurestyle{<value>}** and **\fontfigurealignment{<value>}**.

If you want to change the font family without changing the figure version, use **\fontbasefamily{<value>}**. (All these commands require a succeeding **\selectfont** to make the changes take effect, just as the standard NFSS axes do.)

For choosing the figure versions to be used in math mode you can use the corresponding axis *math figure alignment*. Note, there currently is no means for changing the figure style used in math.

## 1.3 Math version

By default, L<sup>A</sup>T<sub>E</sub>X provides two math versions, **normal** and **bold**, as well as commands **\boldmath** and **\unboldmath** for switching between them. fontaxes redefines these commands to operate on the axis *math weight*.

A second axis *math figure alignment* is introduced that allows you to switch between **tabular** and **proportional** figures using **\tabularmath** and **\proportionalmath**. (This assumes the presence of additional math versions

<sup>1</sup>Still lacking better names.

<sup>2</sup>We are planning to encode the figure version in the font shape instead.

command	axis	value
<code>\upshape</code>	<code>\fontprimaryshape</code>	<code>\updefault</code>
<code>\itshape</code>		<code>\itdefault</code>
<code>\slshape</code>		<code>\sldefault</code>
<code>\swshape</code>		<code>\swdefault</code>
<code>\ulcshape</code>	<code>\fontsecondaryshape</code>	<code>\ulcdefault</code>
<code>\scshape</code>		<code>\scdefault</code>
<code>\sscshape</code>		<code>\sscdefault</code>
<code>\txfigures</code>	<code>\fontfigurestyle</code>	text
<code>\lnfigures</code>		lining
<code>\tbfigures</code>	<code>\fontfigurealignment</code>	tabular
<code>\prfigures</code>		proportional
—	<code>\fontbasefamily</code>	
<code>\boldmath</code>	<code>\mathweight</code>	bold
<code>\unboldmath</code>		normal
<code>\tabularmath</code>	<code>\mathfigurealignment</code>	tabular
<code>\proportionalmath</code>		proportional

Table 1: Author commands set values on axes

tabular and boldtabular. fontaxes will copy the setups of math versions **normal** and **bold** at the end of the preamble in case you do not provide your own declarations.)

`\mathweight` You can directly assign values to the axes using the low-level commands  
`\mathfigurealignment` `\mathweight{⟨value⟩}` and `\mathfigurealignment{⟨value⟩}`.

Table 1 summarizes which commands set which values on which axes.

## 1.4 Additional commands

`\textsw` Similar to the well-known `\textit`, `\textsc`, etc., this package provides the follow-  
`\textssc` ing commands that apply the font change to their argument only. For example,  
`\textulc` `\textsw{⟨text⟩}` is roughly equivalent to `{\swshape ⟨text⟩}` (but automatically  
`\textfigures` adds italic corrections).  
`\liningfigures`  
`\tabularfigures` command corresponding switch(es)  
`\proportionalfigures` `\textsw` `\swshape`  
`\textssc` `\sscshape`  
`\textulc` `\ulcshape`  
`\textfigures` `\txfigures`  
`\liningfigures` `\lnfigures`  
`\tabularfigures` `\tbfigures` `\tabularmath`  
`\proportionalfigures` `\prfigures` `\proportionalmath`

`\figureversion` The command `\figureversion{<options>}` allows easy switching of multiple aspects of figures simultaneously. It takes as an argument a comma-separated list of one or more of the following options:

option	effect
text, osf	<code>\txfigures</code>
lining, lf	<code>\lnfigures</code>
tabular, tab	<code>\tbfigures</code> <code>\tabularmath</code>
proportional, prop	<code>\prfigures</code> <code>\proportionalmath</code>

## 2 Naming conventions

How to name your font families and shapes so they will work with this package.  
(To be done ...)

## 3 Implementation

### 3.1 High-level author commands (Level 1)

#### 3.1.1 Shape

```

\upshape Axis 1: primary shape
\itshape 1 {*package}
\slshape 2 \DeclareRobustCommand\upshape{\not@math@alphabet\upshape\relax
\swshape 3 \fontprimaryshape\updefault\selectfont}
          4 \DeclareRobustCommand\itshape{\not@math@alphabet\itshape\mathit
          5 \fontprimaryshape\itdefault\selectfont}
          6 \DeclareRobustCommand\slshape{\not@math@alphabet\slshape\relax
          7 \fontprimaryshape\sldefault\selectfont}
          8 \DeclareRobustCommand\swshape{\not@math@alphabet\swshape\relax
          9 \fontprimaryshape\swdefault\selectfont}

\scshape Axis 2: secondary shape
\sscshape 10 \DeclareRobustCommand\scshape{\not@math@alphabet\scshape\relax
\ulcshape 11 \fontsecondaryshape\scdefault\selectfont}
          12 \DeclareRobustCommand\sscshape{\not@math@alphabet\sscshape\relax
          13 \fontsecondaryshape\sscdefault\selectfont}
          14 \DeclareRobustCommand\ulcshape{\not@math@alphabet\ulcshape\relax
          15 \fontsecondaryshape\ulcdefault\selectfont}

\swdefault
\ulcdefault 16 \providecommand\swdefault{sw}
\sscdefault 17 \providecommand\ulcdefault{ulc}
          18 \providecommand\sscdefault{ssc}

\textsw
\textssc 19 \DeclareTextFontCommand{\textsw}{\swshape}
\textulc

```

```

20 \DeclareTextFontCommand{\textssc}{\sscshape}
21 \DeclareTextFontCommand{\textulc}{\ulcshape}

```

### 3.1.2 Figure version

```

\txfigures Axis 1: figure style
\lnfigures 22 \def\txfigures{\@nomath\txfigures
23 \fontfigurestyle{text}\selectfont}
24 \def\lnfigures{\@nomath\lnfigures
25 \fontfigurestyle{lining}\selectfont}

\tbfigures Axis 2: figure alignment
\prfigures 26 \def\tbfigures{\@nomath\tbfigures
27 \fontfigurealignment{tabular}\selectfont}
28 \def\prfigures{\@nomath\prfigures
29 \fontfigurealignment{proportional}\selectfont}

\figureversion This code originally appeared in the package MinionPro. I have adapted it to work
within fontaxes' framework and also changed some option names.
30 \newcommand\fa@fv@prefix{\fa@fv@switch@}
31 \newcommand*\fa@fv@newoption[1]
32 {\expandafter\newcommand\csname\fa@fv@prefix #1\endcsname}
33 \fa@fv@newoption{text} {\txfigures}
34 \fa@fv@newoption{osf} {\txfigures}
35 \fa@fv@newoption{lining} {\lnfigures}
36 \fa@fv@newoption{lf} {\lnfigures}
37 \fa@fv@newoption{tabular} {\tbfigures\tabularmath}
38 \fa@fv@newoption{tab} {\tbfigures\tabularmath}
39 \fa@fv@newoption{proportional}{\prfigures\proportionalmath}
40 \fa@fv@newoption{prop} {\prfigures\proportionalmath}

We simply iterate over the list of figure versions specified in the argument to
\figureversion and check if we have specified a matching option.
41 \newcommand\fa@fv@list{}
42 \newcommand\fa@fv{}
43 \DeclareRobustCommand*\figureversion[1]{%
44 \edef\fa@fv@list{\zap@space#1 \@empty}%
45 \@for\fa@fv:=\fa@fv@list\do{%
46 \ifundefined{\fa@fv@prefix\fa@fv}{%
47 \PackageWarning{fontaxes}%
48 {Unknown figure style '\fa@fv'\MessageBreak
49 specified as the argument to \string\figureversion.\MessageBreak
50 Figure style not changed}%
51 }{%
52 \@nameuse{\fa@fv@prefix\fa@fv}%
53 }%
54 }%
55 }

```

We have made `\figureversion` robust to protect it in moving arguments (e.g., section titles). Additionally, we want it to simply be ignored when `hyperref` is building PDF strings (e.g., for bookmarks). The same is true for similar commands, but we only include a selection of them (only the forms with arguments).

```

56 \AtBeginDocument{
57   \ifpackageloaded{hyperref}{%
58     \pdfstringdefDisableCommands{%
59       \let\figureversion\@gobble
60       \let\textfigures\@firstofone
61       \let\liningfigures\@firstofone
62       \let\tabularfigures\@firstofone
63       \let\proportionalfigures\@firstofone
64       \let\textsw\@firstofone
65       \let\textssc\@firstofone
66       \let\textulc\@firstofone
67     }%
68   }{}%
69 }

```

Axis 3: base family `\fontbasefamily{...}`

```

\textfigures
\liningfigures 70 \DeclareTextFontCommand{\textfigures}{\txfigures}
\tabularfigures 71 \DeclareTextFontCommand{\liningfigures}{\lnfigures}
\proportionalfigures 72 \DeclareTextFontCommand{\tabularfigures}{\tbfigures\tabularmath}
73 \DeclareTextFontCommand{\proportionalfigures}
74 {\prfigures\proportionalmath}

```

### 3.1.3 Math version

```

\boldmath Axis 1: weight
\unboldmath 75 \def\boldmath{\@nomath\boldmath
76   \mathweight{bold}}
77 \def\unboldmath{\@nomath\unboldmath
78   \mathweight{normal}}

```

```

\tabularmath Axis 2: figure alignment
\proportionalmath 79 \def\tabularmath{\@nomath\tabularmath
80   \mathfigurealignment{tabular}}
81 \def\proportionalmath{\@nomath\proportionalmath
82   \mathfigurealignment{proportional}}

```

## 3.2 Low-level author commands (Level 2)

```

\mathweight{bold,normal} sets \mathversion
\mathfigurealignment{tabular,proportional} sets \mathversion
\fontfigurestyle{text,lining} sets \fontfamily
\fontfigurealignment{tabular,proportional} sets \fontfamily

```

```

\fontbasefamily{...} sets \fontfamily
\fontprimaryshape{n,it,sl,sw} sets \fontshape
\fontsecondaryshape{ulc,sc,ssc} sets \fontshape

\mathweight
\mathfigurealignment 83 \DeclareRobustCommand\mathweight[1]{%
84 \fa@get@math \edef\fa@math@weight{#1}\fa@set@math}
85 \DeclareRobustCommand\mathfigurealignment[1]{%
86 \fa@get@math \edef\fa@math@align{#1}\fa@set@math}

\fontfigurestyle
\fontfigurealignment 87 \DeclareRobustCommand\fontfigurestyle[1]{%
\fontbasefamily 88 \fa@get@family \edef\fa@figure@style{#1}\fa@set@family}
89 \DeclareRobustCommand\fontfigurealignment[1]{%
90 \fa@get@family \edef\fa@figure@align{#1}\fa@set@family}
91 \DeclareRobustCommand\fontbasefamily[1]{%
92 \fa@get@family \edef\fa@family@base{#1}\fa@set@family}

\fontprimaryshape
\fontsecondaryshape 93 \DeclareRobustCommand\fontprimaryshape[1]{%
94 \fa@get@shape \edef\fa@shape@one{#1}\fa@set@shape}
95 \DeclareRobustCommand\fontsecondaryshape[1]{%
96 \fa@get@shape \edef\fa@shape@two{#1}\fa@set@shape}

```

### 3.3 Internals (Layer 3)

```

\fa@set@math sets \mathversion
\fa@set@family sets \fontfamily
\fa@set@shape sets \fontshape

\fa@math@weight The macros that hold the current values of the axes (here with some default values
\fa@math@align that will most certainly be overwritten during initialization; see \fa@get@...)
\fa@family@base 97 \newcommand*\fa@math@weight{normal}
\fa@figure@style 98 \newcommand*\fa@math@align{proportional}
\fa@figure@align 99 \newcommand*\fa@family@base{MinionPro}
\fa@shape@one 100 \newcommand*\fa@figure@style{text}
\fa@shape@two 101 \newcommand*\fa@figure@align{proportional}
102 \newcommand*\fa@shape@one{n}
103 \newcommand*\fa@shape@two{ulc}

\fa@set@math
\fa@set@family 104 \newcommand*\fa@set@math{%
\fa@set@shape 105 \fa@encode@math
106 \mathversion{\fa@code}%
107 \fa@save\math@version}
108 \newcommand*\fa@set@family{%
109 \fa@encode@family
110 \fontfamily{\fa@code}%

```

```

111 \fa@save\f@family}
112 \newcommand*\fa@set@shape{%
113 \fa@encode@shape
114 \fontshape{\fa@code}%
115 \fa@save\f@shape}

\fa@get@math Check for changes: if changed, try to decode and update axes.
\fa@get@family 116 \newcommand*\fa@get@math{%
\fa@get@shape 117 \iffa@changed\math@version{%
118 \fa@decode@{math}{\math@version}%
119 \ifx\fa@edoc\relax\else
120 \edef\fa@math@weight{\expandafter\@firstoftwo\fa@edoc}%
121 \edef\fa@math@align{\expandafter\@secondoftwo\fa@edoc}%
122 \fi
123 \fa@save\math@version
124 }{}}%
125 }

126 \newcommand*\fa@get@family{%
127 \iffa@changed\f@family{%
128 \let\fa@edoc\relax
129 \expandafter\fa@split@family\f@family--\@nnil
130 \ifx\fa@split@suffix\relax\else
131 \fa@decode@{figures}{\fa@split@suffix}%
132 \fi
133 \ifx\fa@edoc\relax

Try alternative
134 \expandafter\fa@split@familyalt\f@family
135 \@empty\@empty\@empty\@empty\@nnil
136 \ifx\fa@split@suffix\relax\else
137 \fa@decode@{figuresalt}{\fa@split@suffix}%
138 \fi
139 \ifx\fa@edoc\relax
140 \fa@warn@undecodable{family '\f@family'}%
141 \edef\fa@family@base{\f@family}%
142 \else
143 \edef\fa@family@base{\fa@split@prefix}%
144 \edef\fa@figure@style{\expandafter\@firstoftwo\fa@edoc}%

Do not overwrite align (does not occur in alternative naming scheme)
145 \fi
146 \else

Store values
147 \edef\fa@family@base{\fa@split@prefix}%
148 \edef\fa@figure@style{\expandafter\@firstoftwo\fa@edoc}%
149 \edef\fa@figure@align{\expandafter\@secondoftwo\fa@edoc}%
150 \fi
151 }{}}%
152 }

```



```

153 \newcommand*\fa@get@shape{%
154   \iffa@changed\f@shape{%
155     \fa@decode@{shape}{\f@shape}%
156     \ifx\fa@edoc\relax\else
157       \edef\fa@shape@one{\expandafter\@firstoftwo\fa@edoc}%
158       \edef\fa@shape@two{\expandafter\@secondoftwo\fa@edoc}%
159     \fi
160     \fa@save\f@shape
161   }{}%
162 }

```

### 3.4 Encoding

```

\fa@encode@math
\fa@encode@family 163 \newcommand*\fa@encode@math{%
\fa@encode@figures 164 \fa@encode@{math}{\fa@math@weight}{\fa@math@align}}%
\fa@encode@figuresalt 165 }
\fa@encode@shape Default is concatenation

166 \newcommand*\fa@encode@math@default{%
167   \edef\fa@code{\fa@math@weight\fa@math@align}}

168 \newcommand*\fa@encode@family{%
169   \fa@encode@{family}
170   {\fa@family@base}{\fa@figure@style}{\fa@figure@align}}%
171 }

Try different naming conventions

172 \newcommand*\fa@encode@family@default{%
173   \fa@encode@figures
174   \edef\fa@code{\fa@family@base-\fa@code}%
175   \fa@check@family\fa@code
176   \iffa@exists\else
177     \fa@encode@figuresalt
178     \edef\fa@code{\fa@family@base\fa@code}%
179     \fa@check@family\fa@code
180     \iffa@exists\else
181       \edef\fa@code{\fa@family@base}%
182     \fi
183   \fi
184 }

185 \newcommand*\fa@encode@figures{%
186   \fa@encode@{figures}{\fa@figure@style}{\fa@figure@align}}%
187 }

188 \newcommand*\fa@encode@figures@default{%
189   \edef\fa@code{0sF}%
190   \PackageWarning{fontaxes}{Unknown figure version
191     '\fa@figure@style\space + \fa@figure@align'\MessageBreak
192     Encoding to '\fa@code'}%
193 }

```

```

194 \newcommand*\fa@encode@figuresalt{%
195   \fa@encode@{figuresalt}{\fa@figure@style}\fa@figure@align}}%
196 }
197 \newcommand*\fa@encode@figuresalt@default{%
198   \PackageWarning{fontaxes}{Unknown figure version
199     '\fa@figure@style\space + \fa@figure@align'\MessageBreak
200     Encoding to '\fa@code'}%
201   \edef\fa@code{j}%
202 }

203 \newcommand*\fa@encode@shape{%
204   \fa@encode@{shape}{\fa@shape@one}\fa@shape@two}}%
205 }

Default is (reverse) concatenation
206 \newcommand*\fa@encode@shape@default{%
207   \edef\fa@code{\fa@shape@two\fa@shape@one}%
208 }

```

\fa@encode@

```

209 \newcommand*\fa@encode@[2]{%
210   \ifundefined{fa@encode@#1#2}
211     {\@nameuse{fa@encode@#1@default}}
212     {\edef\fa@code{\@nameuse{fa@encode@#1#2}}}%
213 }

```

\fa@naming@exception To do: Add an user interface to specifying naming exceptions

```

214 \newcommand*\fa@naming@exception[3]{%
215   \expandafter\edef\csname fa@encode@#1#2\endcsname{#3}%
216 }

```

The defaults n and ulc disappear when combined.

```

217 \fa@naming@exception{shape}{\n}{ulc}{n}
218 \fa@naming@exception{shape}{\n}{sc}{sc}
219 \fa@naming@exception{shape}{\n}{ssc}{ssc}
220 \fa@naming@exception{shape}{\it}{ulc}{it}
221 \fa@naming@exception{shape}{\sl}{ulc}{sl}
222 \fa@naming@exception{shape}{\sw}{ulc}{sw}

```

The defaults disappear in the concatenation. boldtabular is formed regularly.

```

223 \fa@naming@exception{math}{\normal}{proportional}{normal}
224 \fa@naming@exception{math}{\normal}{tabular}{tabular}
225 \fa@naming@exception{math}{\bold}{proportional}{bold}

```

Provide abbreviations for font family suffixes.

```

226 \fa@naming@exception{figures}{\text}{proportional}{0sF}
227 \fa@naming@exception{figures}{\text}{tabular}{T0sF}
228 \fa@naming@exception{figures}{\lining}{proportional}{LF}
229 \fa@naming@exception{figures}{\lining}{tabular}{TLF}

```

The j/x naming convention does not know about different figure alignments. Let us silently ignore these.

```

230 \fa@naming@exception{figuresalt}{\text}{proportional}}{j}
231 \fa@naming@exception{figuresalt}{\text}{tabular}}{j}
232 \fa@naming@exception{figuresalt}{\lining}{proportional}}{x}
233 \fa@naming@exception{figuresalt}{\lining}{tabular}}{x}

```

### 3.5 Decoding

Detect if `\mathversion`, `\fontshape`, `\fontfamily` have been used not under control of this package.

`\fa@figure@style@domain` Assuming a injective encoding function, we can construct decoding tables when  
`\fa@figure@align@domain` we know the function's domain. To do: Warn if decoding entries are overwritten  
`\fa@shape@one@domain` (if the function is not injective).

```

\fa@shape@two@domain 234 \newcommand*\fa@figure@style@domain{text,lining}
\fa@math@weight@domain 235 \newcommand*\fa@figure@align@domain{proportional,tabular}
\fa@math@align@domain 236 \newcommand*\fa@shape@one@domain{n,it,sl,sw}
237 \newcommand*\fa@shape@two@domain{ulc,sc,ssc}
238 \newcommand*\fa@math@weight@domain{normal,bold}
239 \newcommand*\fa@math@align@domain{proportional,tabular}

```

```

\fa@create@decode@table #1 name, #2 list of axes
240 \newcommand*\fa@create@decode@table[2]{%
241   \begin{group}
242   \fa@foreach{#2}{%
243     \@nameuse{fa@encode@#1}%
244     \global\expandafter
245     \edef\csname fa@decode@#1{\fa@code}\endcsname{#2}%
246   }%
247   \endgroup
248 }
249 \AtEndOfPackage{
250   \fa@create@decode@table{figures}
251   {\fa@figure@style}{\fa@figure@align}}
252 \fa@create@decode@table{figuresalt}
253   {\fa@figure@style}{\fa@figure@align}}
254 \fa@create@decode@table{shape}
255   {\fa@shape@one}{\fa@shape@two}}
256 \fa@create@decode@table{math}
257   {\fa@math@weight}{\fa@math@align}}
258 }

```

```

\fa@warn@undecodable
259 \newcommand*\fa@warn@undecodable[1]{%
260   \PackageWarning{fontaxes}{I don't know how to decode\MessageBreak #1}}

```

`\fa@decode@` Interpret the decoding tables.  
261 \newcommand\*\fa@decode@[2]{%

```

262 \ifundefined{fa@decode@#1{#2}}{%
263   \let\fa@edoc\relax
264   \fa@warn@undecodable{#1 ' #2'}%
265 }{\edef\fa@edoc{\@nameuse{fa@decode@#1{#2}}}%
266 }

```

```

\fa@save Save states of macros for future comparison
\iffa@changed
267 \newcommand*\iffa@changed[1]{%
268   \expandafter\ifx\csname fa@last@\string#1\endcsname#1%
269   \expandafter\@secondoftwo
270   \else
271   \expandafter\@firstoftwo
272   \fi
273 }
274 \newcommand*\fa@save[1]{%
275   \expandafter\let\csname fa@last@\string#1\endcsname#1%
276 }

```

### 3.6 Compatibility

If no math versions `tabular` and `boldtabular` are defined in the preamble, we provide defaults by copying the states of `normal` and `bold` (assuming, in turn, that these two exist).

```

277 \AtBeginDocument{%
278   \fa@provide@mv@copy{tabular}{normal}%
279   \fa@provide@mv@copy{boldtabular}{bold}%
280 }

```

`\fa@provide@mv@copy` Declare math version `#1` to be a copy of math version `#2` if `#1` does not exist already. To accomplish this we have to know that a math version's configuration is basically stored in a macro `\mv@<name>` (which makes us dependent on the NFSS implementation; sigh ...).

```

281 \newcommand*\fa@provide@mv@copy[2]{%
282   \ifundefined{mv@#1}{%
283     \DeclareMathVersion{#1}%
284     \expandafter\let\csname mv@#1\expandafter\endcsname
285     \csname mv@#2\endcsname
286   }{%
287 }

```

### 3.7 Tools

`\fa@check@family` Check if family switching would yield an existing shape.

```

\iffa@exists
288 \newif\iffa@exists
289 \newcommand*\fa@check@family[1]{%
290   \begingroup
291   \fontfamily{#1}\try@load@fontshape
292   \expandafter

```

```

293 \ifx\csname\curr@fontshape\endcsname\relax
294 \aftergroup\fa@existsfalse
295 \else
296 \aftergroup\fa@existstrue
297 \fi
298 \endgroup
299 }

\fa@split@prefix The results of splitting a family name.
\fa@split@suffix 300 \newcommand*\fa@split@prefix{}
301 \newcommand*\fa@split@suffix{}

\fa@split@family Font name contains one hyphen, split there
302 \newcommand*\fa@split@family{}
303 \def\fa@split@family#1-#2-#3\@nnil{%
304 \let\fa@split@prefix\relax
305 \let\fa@split@suffix\relax
306 \def\@tempa{#3}%
307 \ifx\@tempa\@empty\else
308 \def\fa@split@suffix{#2}%
309 \ifx\fa@split@suffix\@empty
310 \let\fa@split@suffix\relax
311 \else
312 \def\fa@split@prefix{#1}%
313 \fi
314 \fi
315 }

\fa@split@familyalt Name consists of four characters, split off the last one
316 \newcommand*\fa@split@familyalt{}
317 \def\fa@split@familyalt#1#2#3#4#5\@nnil{%
318 \let\fa@split@prefix\relax
319 \let\fa@split@suffix\relax
320 \edef\@tempa{#5}%
321 \ifx\@tempa\@empty
322 \ifx\@empty#4\else
323 \def\fa@split@prefix{#1#2#3}%
324 \def\fa@split@suffix{#4}%
325 \fi
326 \fi
327 }

\fa@foreach Execute #2 for each combination of values of the axes given in #1 (in the form
{\cs}{\cs}...).
328 \newcommand\fa@foreach[2]{%
329 \begingroup
330 \def\fa@foreach@{#2}%
331 \@tfor\@tempa:=#1\do{%
332 \@temptokena\expandafter{\fa@foreach@}%

```

```

333     \edef\fa@foreach@{%
334         \noexpand\@for
335         \expandafter\noexpand\@tempa:=%
336         \expandafter\noexpand\csname
337             \expandafter\expandafter
338             \expandafter\@gobble
339             \expandafter\string\@tempa
340             @domain%
341         \endcsname
342         \noexpand\do{\the\@temptokena}%
343     }%
344 }%
345 \expandafter\endgroup\fa@foreach@
346 }
347 \end{package}

```

### 3.8 Tests

The file `test-fontaxes.tex` (docstrip target `test`) exercises some features of `fontaxes`. Since it is rather ad-hoc code, it is not shown here. (It also requires the package `MinionPro`.)