The `microtype` package

Subliminal refinements towards typographical perfection

— IMPLEMENTATION —

R Schlicht   v3.0b

w.m.l@gmx.net   2021/12/10

https://github.com/schlcht/microtype

The `microtype` package provides a LaTeX interface to the micro-typographic extensions that were introduced by pdfTeX and have since also propagated to LuaTeX and XeTeX: most prominently, character protrusion and font expansion, furthermore the adjustment of interword spacing and additional kerning, as well as hyphenatable letterspacing (tracking) and the possibility to disable all or selected ligatures. These features may be applied to customisable sets of fonts, and all micro-typographic aspects of the fonts can be configured in a straight-forward and flexible way. Settings for various fonts are provided.

Note that character protrusion requires pdfTeX (version 0.14f or later), LuaTeX, or XeTeX (at least version 0.9997). Font expansion works with pdfTeX (version 1.20 for automatic expansion) or LuaTeX. The package will by default enable protrusion and expansion if they can safely be assumed to work. Disabling ligatures requires pdfTeX (≥ 1.30) or LuaTeX, while the adjustment of interword spacing and of kerning only works with pdfTeX (≥ 1.40). Letterspacing is available with pdfTeX (≥ 1.40) or LuaTeX (≥ 0.62).

The alternative package `letterspace`, which also works with plain TeX, provides the user commands for letterspacing only, omitting support for all other extensions (see section 7 of the User manual).

Contents

List of Tables

List of Figures

User manual (external document)

1 Implementation

The `docstrip` modules in this file are:

driver: The documentation driver, only visible in the `dtx` file.

package: The code for the `microtype` package (`microtype.sty`).

show: The code for the `microtype-show` package (`microtype-show.sty`).

pdf-: Definitions specific to pdfTEX (`microtype-pdftex.def`).

lua-: Definitions specific to LuaTEX (`microtype-luatex.def`).

xe-: Definitions specific to XƎTEX (`microtype-xetex.def`).

letterspace: The code for the `letterspace` package (`letterspace.sty`).

> plain: Code for `eplain`, `miniltx` (`letterspace` only).

debug: Code for additional output in the log file.
Used for – surprise! – debugging purposes.

luafile: Lua functions (`microtype.lua`).

config: Surrounds all configuration modules.

> cfg-t: Surrounds (Latin) text configurations.
>
>> m-t: The main configuration file (`microtype.cfg`).
>> bch: Settings for Bitstream Charter (`mt-bch.cfg`).
>> blg: Settings for Bitstream Letter Gothic (`mt-blg.cfg`).
>> cmr: Settings for Computer Modern Roman (`mt-cmr.cfg`).
>> ebg: Settings for EB Garamond (`mt-EBGaramond.cfg`).
>> ppl: Settings for Palatino (`mt-ppl.cfg`).
>> ptm: Settings for Times (`mt-ptm.cfg`).
>> pmn: Settings for Adobe Minion (`mt-pmn.cfg`).
>> Contributed by *Harald Harders*.
>> ugm: Settings for URW Garamond (`mt-ugm.cfg`).
>
> cfg-u: Surrounds non-text configurations (U encoding).
>
>> msa: Settings for AMS 'a' symbol font (`mt-msa.cfg`).
>> msb: Settings for AMS 'b' symbol font (`mt-msb.cfg`).
>> euf: Settings for Euler Fraktur font (`mt-euf.cfg`).
>> eur: Settings for Euler Roman font (`mt-eur.cfg`).
>> eus: Settings for Euler Script font (`mt-eus.cfg`).
>
> cfg-e: Surrounds Euro symbol configurations.
>
>> zpeu: Settings for Adobe Euro symbol fonts (`mt-zpeu.cfg`).
>> mvs: Settings for `marvosym` Euro symbol (`mt-mvs.cfg`).

test: A helper file that may be used to create and test protrusion settings (`test-microtype.tex`).

And now for something completely different.

1 ⟨*package | letterspace⟩

1.1 Preliminaries

`\MT@MT`   This is us.

```
2 \def\MT@MT
3 ⟨package⟩    {microtype}
4 ⟨letterspace⟩  {letterspace}
```

`\MT@fix@catcode`   We have to make sure that the category codes of some characters are correct (the german package, for instance, makes " active). Probably overly cautious. Ceterum censeo: it should be forbidden for packages to change catcodes within the preamble.

`\MT@restore@catcodes`   Polite as we are, we'll restore them afterwards.

```
5 \let\MT@restore@catcodes\@empty

6 \def\MT@fix@catcode#1#2{%
7   \edef\MT@restore@catcodes{%
8     \MT@restore@catcodes
9     \catcode#1=\the\catcode#1\relax
10   }%
11   \catcode#1=#2\relax
12 }
13 \MT@fix@catcode{17}{14}% ^^Q (comment)
14 \MT@fix@catcode{24} {9}% ^^X (ignore)
15 ⟨package⟩\MT@fix@catcode{33}{12}% !
16 ⟨package⟩\MT@fix@catcode{34}{12}% "
17 \MT@fix@catcode{36} {3}% $ (math shift)
18 \MT@fix@catcode{39}{12}% '
19 \MT@fix@catcode{42}{12}% *
20 \MT@fix@catcode{43}{12}% +
21 \MT@fix@catcode{44}{12}% ,
22 \MT@fix@catcode{45}{12}% -
23 \MT@fix@catcode{58}{12}% :
24 \MT@fix@catcode{60}{12}% <
25 \MT@fix@catcode{61}{12}% =
26 \MT@fix@catcode{62}{12}% >
27 ⟨package⟩\MT@fix@catcode{63}{12}% ?
28 \MT@fix@catcode{94} {7}% ^ (superscript)
29 \MT@fix@catcode{96}{12}% `
30 ⟨package⟩\MT@fix@catcode{124}{12}% |
```

These are all commands for the outside world. We define them here as blank commands, so that they won't generate an error if we are not running pdfTeX.

```
31 ⟨*package⟩
32 \newcommand*\DeclareMicrotypeSet[3][]{}
33 \newcommand*\UseMicrotypeSet[2][]{}
34 \newcommand*\DeclareMicrotypeSetDefault[2][]{}
35 \newcommand*\SetProtrusion[3][]{}
36 \newcommand*\SetExpansion[3][]{}
37 \newcommand*\SetTracking[3][]{}
38 \newcommand*\SetExtraKerning[3][]{}
39 \newcommand*\SetExtraSpacing[3][]{}
40 \newcommand*\DisableLigatures[2][]{}
41 \newcommand*\DeclareCharacterInheritance[3][]{}
42 \newcommand*\DeclareMicrotypeVariants[1]{}
43 \newcommand*\DeclareMicrotypeAlias[2]{}
44 \newcommand*\LoadMicrotypeFile[1]{}
45 \newcommand*\DeclareMicrotypeBabelHook[2]{}
46 \newcommand*\microtypesetup[1]{}
47 \newcommand*\microtypecontext[1]{}
48 \newcommand*\textmicrotypecontext[2]{#2}
49 \newcommand\leftprotrusion[1]{#1}
50 \newcommand\rightprotrusion[1]{#1}
51 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
52 ⟨/package⟩
53 \newcommand*\lsstyle{}
```

```
54 \newcommand\textls[2][]{}
55 \def\textls#1#{}
56 \newcommand*\lslig[1]{#1}
57 ⟨*package⟩
58 }
```

These commands also have a starred version.

```
59 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
60 \def\DeclareMicrotypeVariants#1#{\@gobble}
```

Set declarations are only allowed in the preamble (resp. the main configuration file). The configuration commands, on the other hand, must be allowed in the document, too, since they may be called inside font configuration files, which, in principle, may be loaded at any time.

```
61 \@onlypreamble\DeclareMicrotypeSet
62 \@onlypreamble\UseMicrotypeSet
63 \@onlypreamble\DeclareMicrotypeSetDefault
64 \@onlypreamble\DisableLigatures
65 \@onlypreamble\DeclareMicrotypeVariants
66 \@onlypreamble\DeclareMicrotypeBabelHook
```

Don't load `letterspace`.

```
67 \expandafter\let\csname ver@letterspace.sty\endcsname\@empty
```

`\MT@old@cmd`   The old command names had one more hunch (`\..MicroType..`). Before finally letting them sink into oblivion, raise an error.

```
68 \def\MT@old@cmd#1#2{%
69   \newcommand*#1{\MT@error{%
70     \string#1 is deprecated. Please use\MessageBreak
71     \string#2 instead}{As I said}%
72     \let #1#2#2}}
73 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
74 \MT@old@cmd\DeclareMicroTypeSet   \DeclareMicrotypeSet
75 \MT@old@cmd\UseMicroTypeSet       \UseMicrotypeSet
76 \MT@old@cmd\LoadMicroTypeFile     \LoadMicrotypeFile
77 ⟨/package⟩
```

`\MT@warning`
`\MT@warning@nl`   Communicate.
`\MT@info`
`\MT@info@nl`
`\MT@vinfo`
`\MT@error`
`\MT@warn@err`

```
78 \def\MT@warning{\PackageWarning\MT@MT}
79 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
80 ⟨*package⟩
81 \def\MT@info{\PackageInfo\MT@MT}
82 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
83 \let\MT@vinfo\@gobble
84 \def\MT@error{\PackageError\MT@MT}
85 \def\MT@warn@err#1{\MT@error{#1}{%
86   This error message appears because you loaded the `\MT@MT'\MessageBreak
87   package with the option `verbose=errors'. Consult the documentation\MessageBreak
88   in \MT@MT.pdf to find out what went wrong.}}
```

1.1.1 Debugging

`\tracingmicrotype`   Cases for `\tracingmicrotype`:
`\MT@dinfo`
`\MT@dinfo@nl`   0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```
89  ⟨*debug⟩
90  \MT@warning@nl{This is the debug version}
91  \newcount\tracingmicrotype
92  \tracingmicrotype=2
93  \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
94  \def\MT@info@nl#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
95  \let\MT@vinfo\MT@info@nl
96  \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
97  \def\MT@warning@nl#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
98  \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
99  \def\MT@dinfo@nl#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@nl{#2}\fi}
```

\tracingmicrotypeinpdf   Another debug method: font switches can be marked in the PDF file with a small
caret, an accompanying popup text box displaying all debug messages.

Cases for \tracingmicrotypeinpdf:

1: show new fonts

2: + show known fonts

```
100  \newcount\tracingmicrotypeinpdf
```

Let's see how it works ... (if you don't see anything special on this page, your PDF
viewer doesn't support annotations).

```
\tracingmicrotypeinpdf=2
```

\MT@pdf@annot   During font setup, we save the text for the popup in \MT@pdf@annot. (This requires
\MT@addto@annot   pdfTEX ≥ 1.30.) The pdftexcmds package provides pdfTEX's utility commands in
\ifMT@inannot   LuaTEX, too.

```
101  \RequirePackage{pdftexcmds}
102  \newif\ifMT@inannot \MT@inannottrue
103  \let\MT@pdf@annot\@empty
104  \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
105    {\def\MessageBreak{^^J\@spaces}%
106     \MT@xadd\MT@pdf@annot{\pdf@escapestring{#1^^J}}}\fi\fi}
```

\iftracingmicrotypeinpdfall   With \tracingmicrotypeinpdfallfalse, the PDF output is (hopefully) identical,
but some font switches will not be displayed; otherwise the output is affected, but
all font switches are visible. In the latter case, we also insert a small kern so that
multiple font switches are discernable.

```
107  \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdfannot   A red caret is shown for fonts which are actually set up by *Microtype*, a green one
marks fonts that we have already seen. The /Caret annotation requires a viewer
for PDF version 1.5 (you could use /Text if you're using an older PDF viewer).

```
108  \def\MT@show@pdfannot#1{%
109    \ifnum\tracingmicrotypeinpdf<#1 \else
110      \iftracingmicrotypeinpdfall\leavevmode\fi
111      \pdfannot height 4pt width 4pt depth 2pt {%
112        /Subtype/Caret
113        /T(\expandafter\string\font@name)
114        \ifcase#1\or
115        /Subj(New font)/C[1 0 0]
116        \else
117        /Subj(Known font)/C[0 1 0]
118        \fi
119        /Contents(\MT@pdf@annot)
120      }%
121      \iftracingmicrotypeinpdfall\kern1pt \fi
122      \global\MT@inannotfalse
123    \fi
124  }
```

```
125 ⟨/debug⟩
126 ⟨/package⟩
127 ⟨/package|letterspace⟩
```

1.1.2 Visual debugging

The microtype-show package offers some tools for preparing protrusion settings. We make use of the microtype infrastructure, redefining some of its internal commands (done later, in sections 1.2.1 and 1.2.7). First, some preparation:

```
128 ⟨*show⟩
129 \RequirePackage{iftex}
130 \ifetex\else
131   \PackageError{microtype-show}
132                {This package only works with e-TeX}{Use e-TeX}
133 \fi
134 \ifxetex
135   \PackageError{microtype-show}
136                {This package only works with pdfTeX or luaTeX}{Don't use XeTeX}
137 \fi
138 \PackageWarning{microtype-show}{DO NOT USE THIS PACKAGE FOR REAL DOCUMENTS\@gobble}
139 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{microtype}}
140 \ProcessOptions\relax
141 \PassOptionsToPackage{verbose}{microtype}
142 \RequirePackage{microtype,graphicx,xcolor}
```

\ifShowGlyphIndex   The following commands are configurable:

\ifShowMissingGlyphs
\GlyphScaleFactor
\Showbaselinecolor
\Showposcolor
\Shownegcolor

```
143 \newif\ifShowGlyphIndex
144 \newif\ifShowMissingGlyphs
145 \newcommand*\GlyphScaleFactor{2}
146 \newcommand*\Showbaselinecolor{\color{black!40}}
147 \newcommand*\Showposcolor{\color{green!50}}
148 \newcommand*\Shownegcolor{\color{red!50}}
```

\MTS@printtext   Make sure to have a readable font.

\MTS@show@index
\MTS@crulefill

```
149 \ifluatex
150   \def\MTS@printtext#1{{\usefont{TU}{lmr}{m}{n}#1}}
151 \else
152   \def\MTS@printtext#1{{\usefont{T1}{cmr}{m}{n}#1}}
153 \fi
154 \def\MTS@show@index#1{\ifShowGlyphIndex{\tiny$_{#1}%
155 %  \ifluatex^{\mathrm{%
156 %    \MT@lua{tex.print(luaotfload.aux.name_of_slot(tonumber([[#1]])))}}}\fi
157   $}\fi\space}
158 \def\MTS@crulefill{\leaders\hrule height \dimexpr1ex/2+.4pt depth -\dimexpr1ex/2\hfill}
```

\MTS@Prot   Add the show commands to microtype's setup.

\MTS@Char

```
159 \g@addto@macro\MT@setupfont{\MTS@Prot\MTS@Char}
160 \let\MTS@Prot\relax
161 \let\MTS@Char\relax
```

\MTS@setup   Common setup. \MTS@glyphlist stores all glyphs we've seen.

\MTS@glyphlist

```
162 \def\MTS@setup{%
163   \fboxsep=0pt
164   \fboxrule=.1pt
165   \raggedright
166   \let\MTS@glyphlist\@gobble
167   \def\MT@feat{pr}%
168 }
```

\ShowProtrusion   Activate the sleeper command, then trigger the setup.

```
169 \newcommand*\ShowProtrusion{%
170   \begingroup
171     \MTS@setup
```

```
172     \let\MTS@Prot\MTS@Prot@do
173     \def\MT@cat{c}%
174     \selectfont
175 }
```

\MTS@Prot@do

But in all other cases of a font being picked up, there should be no special treatment. After we're done, select the previous font again.

```
176 \def\MTS@Prot@do{%
177     \MT@ltx@pickupfont
178     \let\MT@pr@split@val\MTS@pr@split@val
179     \let\MT@load@list\MTS@load@list
180     \let\MT@set@pr@prefixes@\MTS@set@pr@prefixes@
181     \MTS@show@pr
182   \endgroup
183   \aftergroup\selectfont
184 }
```

\ShowCharacterInheritance

```
185 \newcommand*\ShowCharacterInheritance{%
186   \begingroup
187     \MTS@setup
188     \let\MTS@Char\MTS@Char@do
189     \def\MT@cat{inh}%
190     \selectfont
191 }
```

\MTS@Char@do

```
192 \def\MTS@Char@do{%
193     \MT@ltx@pickupfont
194     \let\MT@set@pr@prefixes@\MTS@set@pr@prefixes@
195     \MTS@show@inheritance
196   \endgroup
197   \aftergroup\selectfont
198 }
```

\ShowProtrusionLineGlyph

By glyph.

```
199 \newcommand*\ShowProtrusionLineGlyph[1]{%
200   {\MTS@setup
201    \MTS@showprotrusionline{`#1}}%
202 }
```

\ShowProtrusionLineIndex

By glyph number.

```
203 \newcommand*\ShowProtrusionLineIndex[1]{%
204   {\MTS@setup
205    \MTS@showprotrusionline{#1}}%
206 }
```

\MTS@showprotrusionline
\MTS@lpcode
\MTS@rpcode

```
207 \def\MTS@showprotrusionline#1{%
208   \edef\MTS@lpcode{\number\lpcode\font#1}%
209   \edef\MTS@rpcode{\number\rpcode\font#1}%
210   \char#1%
211     lorem ipsum dolor sit amet, \MTS@crulefill\ %
212     \MTS@printtext{\ifnum\MTS@lpcode=\z@\Showbaselinecolor\fi[\MTS@lpcode]}
213     \fbox{\char#1}\MTS@show@index{\number#1}
214     \MTS@printtext{\ifnum\MTS@rpcode=\z@\Showbaselinecolor\fi[\MTS@rpcode]}
215     \MTS@crulefill\ you know the rest%
216   \char#1\par
217   \ShowDummyLine
218 }
```

\ShowDummyLine

The first and last glyphs in this line should have a straight (non-protruded) shape. We also reset to default shape and series, because that's what, say, italic shapes should be matched with.

```
219 \newcommand*\ShowDummyLine{%
220  {\fontencoding{\encodingdefault}\fontseries{\seriesdefault}\fontshape{\shapedefault}%
221   \selectfont\noindent
222   here is the beginning of a line, \dotfill and here is its end}\par
223 }
```

\ShowProtrusionAll

```
224 \newcommand*\ShowProtrusionAll{%
225  {\MTS@setup
226   \MTS@lede{}%
227   \MT@do@font{\iffontchar\font\@tempcnta\MTS@showprotrusionline{\@tempcnta}\fi}}%
228 }
```

\ShowProtrusionDefined

```
229 \newcommand*\ShowProtrusionDefined{%
230   {\MTS@setup
231    \MTS@lede{defined}%
232    \let\MTS@first\@gobble
233    \let\MTS@second\@firstofone
234    \MT@do@font{%
235      \MTS@firstorsecond
236      \MTS@temp{%
237        \iffontchar\font\@tempcnta\MTS@showprotrusionline{\@tempcnta}\else
238          \MT@warning@nl{Glyph \the\@tempcnta\space is missing in font
239                        \MessageBreak\font@name}%
240      \fi}}}%
241 }
```

\ShowProtrusionMissing

```
242 \newcommand*\ShowProtrusionMissing{%
243   {\MTS@setup
244    \MTS@lede{missing}%
245    \let\MTS@first\@firstofone
246    \let\MTS@second\@gobble
247    \MT@do@font{%
248      \MTS@firstorsecond
249      \iffontchar\font\@tempcnta\MTS@temp{\MTS@showprotrusionline{\@tempcnta}}\fi}}%
250 }
```

\MTS@lede

```
251 \def\MTS@lede#1{%
252   \selectfont
253   \edef\MTS@font{\expandafter\string\font@name}%
254   \MTS@printtext{All glyphs \MT@ifempty{#1}{in}{#1 in protrusion list for}
255                 font \texttt{\MTS@font}:}\par
256   \ShowDummyLine
257 }
```

\MTS@firstorsecond

```
258 \def\MTS@firstorsecond{%
259   \let\MTS@temp\MTS@first
260   \ifnum\lpcode\font\@tempcnta=\z@ \else
261     \let\MTS@temp\MTS@second
262   \fi
263   \ifnum\rpcode\font\@tempcnta=\z@ \else
264     \let\MTS@temp\MTS@second
265   \fi
266 }
```

\MTS@charwd
\MTS@lp@   Display the glyph with protrusion.
\MTS@rp@
```
267 \newdimen\MTS@charwd
268 \newdimen\MTS@lp@
269 \newdimen\MTS@rp@
```
\MTS@show@char@pr
```
270 \def\MTS@show@char@pr#1{%
271   \xdef\MTS@glyphlist{\MTS@glyphlist,#1}%
```

```
272    \scalebox{\GlyphScaleFactor}{\strut\escapechar`\\
273      \MTS@charwd=\fontcharwd\MT@font#1\relax
```

The baseline rule.

```
274      {\Showbaselinecolor\vrule width \dimexpr\MTS@charwd+.3em\relax height 1sp depth 0pt}%
275      \hskip-\dimexpr\MTS@charwd+.15em\relax
```

Left protrusion.

```
276      {\ifdim\MTS@lp@<\z@\Shownegcolor\else\Showposcolor\fi
277      \vrule width \ifdim\MTS@lp@<\z@ -\fi\MTS@lp@ height 1em depth .2em}%
278      \hskip\dimexpr\MTS@charwd\ifdim\MTS@lp@>\z@-\MTS@lp@\fi
279                                    \ifdim\MTS@rp@>\z@-\MTS@rp@\fi\relax
```

Right protrusion.

```
280      {\ifdim\MTS@rp@<\z@\Shownegcolor\else\Showposcolor\fi
281      \vrule width \ifdim\MTS@rp@<\z@ -\fi\MTS@rp@ height 1em depth .2em}%
282      \hskip-\dimexpr\MTS@charwd+\fboxrule\ifdim\MTS@rp@<\z@-\MTS@rp@\fi\relax
```

Finally the glyph, so that it's on top.

```
283      \fbox{\char#1}}\,%
284      \MTS@show@index{#1}%
285 }
```

<div style="text-align: right">\MTS@show@char
\MTS@show@char@x</div>

Just show the glyph; the second command also remembers it.

```
286 \def\MTS@show@char#1{\scalebox{\GlyphScaleFactor}{%
287    \strut\fbox{\char#1}}\MTS@show@index{#1}}
288 \def\MTS@show@char@x#1{\xdef\MTS@glyphlist{\MTS@glyphlist,#1}\MTS@show@char{#1}}
```

<div style="text-align: right">\MTS@show@missing</div>

```
289 \def\MTS@show@missing{%
290    \MT@ifdefined@c@T\MT@pr@inh@name{%
291      \MTS@lp@=\z@ \MTS@rp@=\z@
292      \par \MTS@printtext{Glyphs not included in configuration (with defined heirs):}%
293      \MT@do@font{%
294        \edef\MT@temp{\the\@tempcnta}%
295        \MT@ifdefined@n@T{\MT@inh@\MT@pr@inh@name @\MT@temp @}{%
296          \MT@exp@one@n\MT@in@clist\MT@temp\MTS@glyphlist
297          \ifMT@inlist@\else \newline
298          \llap{\MTS@show@char@pr{\MT@temp} \MTS@printtext{=} }%
299           \MT@exp@cs\MT@map@tlist@c
300             {\MT@inh@\MT@pr@inh@name @\the\@tempcnta @}%
301             \MTS@show@char@x
302          \fi
303        }%
304      }%
305    }%
306    \MTS@show@missing@
307 }
```

<div style="text-align: right">\MTS@show@missing@</div>

```
308 \def\MTS@show@missing@{%
309    \par \MTS@printtext{Other glyphs not in configuration:}\newline
310    \MT@do@font{%
311      \edef\MT@temp{\the\@tempcnta}%
312      \MT@exp@one@n\MT@in@clist\MT@temp\MTS@glyphlist
313      \ifMT@inlist@\else
314        \MTS@show@char\MT@temp
315      \fi
316    }%
317 }
```

<div style="text-align: right">\MTS@show@inheritance</div>

```
318 \def\MTS@show@inheritance{%
319    \MT@get@inh@list
320    \MTS@printtext{Character inheritance for font `\texttt{\MT@@font}':}\\
```

```
321 \MT@ifdefined@c@TF\MT@listname{%
322   \MTS@printtext{First matching list is for `\texttt{\@tempa}':\\
323           \texttt{\MT@listname}:}\par\leavevmode
324   \MT@do@font{%
325     \MT@ifdefined@n@T{MT@inh@\MT@listname @\the\@tempcnta @}{%
326       \newline
327       \xdef\MTS@glyphlist{\MTS@glyphlist,\the\@tempcnta}%
328       \llap{\MTS@show@char{\the\@tempcnta}\MTS@printtext{= }}%
329       \MT@exp@cs\MT@map@tlist@c
330         {MT@inh@\MT@listname @\the\@tempcnta @}%
331         \MTS@show@char@x
332     }%
333   }%
334   \MT@ifdefined@n@T{MT@inh@\MT@listname @prefixes}{%
335     \par \MTS@printtext{(with prefixes:)}%
336     \@tempcntb=\z@
337     \let\MTS@show@char@pr\MTS@show@char@x
338     \MT@set@pr@prefixheirs}%
339   \ifShowMissingGlyphs\MTS@show@missing@\fi
340 }{%
341   \MTS@printtext{NOT DEFINED}%
342 }%
343 \par
344 }
345 ⟨/show⟩
```

1.1.3 Requirements

Back to the user packages.

\MT@plain   The letterspace package works with:

0: miniltx

1: eplain

2: LATEX

For plain usage, we have to copy some commands from latex.ltx.

```
346 ⟨*package|letterspace⟩
347 ⟨*plain⟩
348 \def\MT@plain{2}
349 \ifx\documentclass\@undefined
350   \def\MT@plain{1}
351   \def\hmode@bgroup{\leavevmode\bgroup}
352   \def\nfss@text#1{{\mbox{#1}}}
353   \let\@typeset@protect\relax
354   \ifx\eplain\@undefined
355     \def\MT@plain{0}
356     \def\PackageWarning#1#2{%
357       \begingroup
358         \newlinechar=10 %
359         \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
360         \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
361       \endgroup
362     }
363     \def\on@line{ on input line \the\inputlineno}
364     \def\@spaces{\space\space\space\space}
365   \fi
366 \fi
```

\MT@requires@latex   Better use groups than plain ifs.

```
367 \def\MT@requires@latex#1{%
368   \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
369 }
370 ⟨/plain⟩
```

For definitions that depend on e-TeX features.

```
371 \ifcase 0%
372   \ifx\eTeXversion\@undefined 1\else
373     \ifx\eTeXversion\relax     1\else
374       \ifcase\eTeXversion       1\fi
375     \fi
376   \fi
377 \else
378   \catcode`\^^Q=9 \catcode`\^^X=14
379 \fi
380 ⟨letterspace⟩^^Q\MT@warning@nl{This package requires the etex extensions.
381 ⟨letterspace⟩^^Q                \MessageBreak Exiting}\MT@restore@catcodes\endinput
382 ⟨debug⟩\MT@dinfo@nl{0}{this is
383 ⟨debug⟩^^Q not
384 ⟨debug⟩  etex}
```

We check whether we are running pdfTeX, XƎTeX, or LuaTeX, and load the appropriate definition file (later in section 1.4.2).

\MT@clear@options   If we are using neither of these engines, or a too old version, we disable everything and exit.

```
385 \def\MT@clear@options{%
386 ⟨plain⟩  \MT@requires@latex1{%
387   \AtEndOfPackage{\let\@unprocessedoptions\relax\MT@restore@catcodes}%
388   \let\CurrentOption\@empty
389 ⟨plain⟩  }\relax
390 }
```

A hack circumventing the TeX Live 2004 hack which undefines the pdfTeX primitives in the format in order to hide the fact that pdfTeX is being run from the user. This has been *fixed* in TeX Live 2005.

```
391 \ifx\normalpdftexversion\@undefined \else
392   \let\pdftexversion \normalpdftexversion
393   \let\pdftexrevision\normalpdftexrevision
394   \let\pdfoutput      \normalpdfoutput
395 \fi
```

\MT@engine   Old packages might have let \pdftexversion to \relax.

\ifMT@engine@unfit
\MT@engine@minversion
```
396 \let\MT@engine\relax
397 \newif\ifMT@engine@unfit
398 \MT@engine@unfittrue
399 \ifx\pdftexversion\@undefined \else
400   \ifx\pdftexversion\relax \else
401     \def\MT@engine{pdf}
402 ⟨package⟩    \def\MT@engine@minversion{0.14f}
403 ⟨letterspace⟩    \let\MT@pdf@or@lua\@firstoftwo
404     \ifnum\pdftexversion
405 ⟨package⟩        > 13
406 ⟨letterspace⟩        > 139
407       \MT@engine@unfitfalse
408 ⟨package⟩      \ifnum \pdftexversion=14
409 ⟨package⟩        \ifnum \expandafter`\pdftexrevision < `f
410 ⟨package⟩          \MT@engine@unfittrue
411 ⟨package⟩        \fi
412 ⟨package⟩      \fi
413     \fi
414   \fi
415 \fi
416 \ifx\directlua\@undefined \else
417   \ifx\directlua\relax \else
418     \def\MT@engine{lua}
419     \MT@engine@unfitfalse
```

Since approx. LuaTeX 0.80, \pdftexversion is let to \luatexversion, so that we

would be fooled into thinking that pdfTEX is too old.

```
420 ⟨*letterspace⟩
421    \let\MT@pdf@or@lua\@secondoftwo
422    \ifnum\luatexversion < 62 \MT@engine@unfittrue
423    \else
424      \let\MT@lua\directlua
425      \ifnum\luatexversion > 84
426        \let\pdfoutput\outputmode
427        \let\pdfprotrudechars\protrudechars
428        \let\pdfadjustspacing\adjustspacing
429      \fi
430    \fi
431 ⟨/letterspace⟩
432    \fi
433 \fi
434 ⟨*package⟩
435 \ifx\MT@engine\relax
436   \ifx\XeTeXversion\@undefined \else
437     \ifx\XeTeXversion\relax \else
438       \def\MT@engine{xe}
439       \def\MT@engine@minversion{0.9997}
440       \ifdim 0\XeTeXrevision pt > 0.9996pt
441         \MT@engine@unfitfalse
442       \fi
443     \fi
444   \fi
445 \fi
446 ⟨/package⟩
447 ⟨/package|letterspace⟩
```

\MT@pdftex@no   pdfTEX's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdfTEX we're using, if any. \MT@pdftex@no will be used throughout the package to respectively do the right thing. Currently, we have to distinguish the following cases for pdfTEX:

0: not running pdfTEX

1: pdfTEX (< 0.14f) (already checked above)

2: + micro-typographic extensions (0.14f,g)

3: + protrusion relative to 1 em (≥ 0.14h)

4: + automatic font expansion; protrusion no longer has to be set up first; scale factor fixed to 1000; default \efcode = 1000 (≥ 1.20)

5: + \(left,right)marginkern; \pdfnoligatures; \pdfstrcmp; \pdfescapestring (≥ 1.30)

6: + adjustment of interword spacing; extra kerning; \letterspacefont; \pdfmatch[1]; \pdftracingfonts; always e-TEX (≥ 1.40)

7: + \letterspacefont doesn't disable ligatures and kerns; \pdfcopyfont (≥ 1.40.4)

8: + \letterspacefont uses explicit \fontdimen 6 if specified (≥ 1.40.23)

```
448 ⟨*pdf-⟩
449 ⟨debug⟩\MT@dinfo@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
450 \def\MT@pdftex@no{8}
451 \ifnum\pdftexversion = 140
452   \ifnum\pdftexrevision < 23
453     \def\MT@pdftex@no{7}
```

---

1 This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.

```
454     \ifnum\pdftexrevision < 4
455       \def\MT@pdftex@no{6}
456     \fi
457   \fi
458 \else
459   \ifnum\pdftexversion < 140
460     \def\MT@pdftex@no{5}
461     \ifnum\pdftexversion < 130
462       \def\MT@pdftex@no{4}
463       \ifnum\pdftexversion < 120
464         \def\MT@pdftex@no{3}
465         \ifnum\pdftexversion = 14
466           \ifnum \expandafter`\pdftexrevision < `h
467             \def\MT@pdftex@no{2}
468           \fi
469         \fi
470       \fi
471     \fi
472   \fi
473 \fi
```
474 ⟨*debug*⟩\MT@dinfo@nl{0}{pdftex no.: \MT@pdftex@no}
475 ⟨*/pdf-*⟩

\MT@xetex@no   XƎTEX supports character protrusion since version 0.9997. This test is not necessary here, we just keep it for the (unlikely) case that features get added to XƎTEX in the future.

476 ⟨*xe-*⟩
477 ⟨*debug*⟩\MT@dinfo@nl{0}{this is xetex (\the\XeTeXversion\XeTeXrevision)}
478 %\ifdim 0\XeTeXrevision pt < 0.9997pt
479 % \def\MT@xetex@no{1}
480 %\else
481 % \def\MT@xetex@no{2}
482 %\fi
483 ⟨*debug*⟩%\MT@dinfo@nl{0}{xetex no.: \MT@xetex@no}
484 ⟨*/xe-*⟩

\MT@luatex@no   Cases for LuaTEX (\luatexversion ought to have been enabled by the format):

0: N/A

1: LuaTEX (< 0.36)

2: + \directlua without state number (≥ 0.36)

3: + \letterspacefont; non-automatic expansion doesn't work anymore, and automatic expansion in DVI mode is realised by modifying the tracking, not the glyphs[2] (≥ 0.62)

4: + almost all of the pdfTEX primitives have been renamed (≥ 0.85)

5: + default \efcode = 1000; \protrusionboundary [doesn't seem to work] (≥ 0.90)

6: + \glet (≥ 1.10)

Also, sometime between 1.0.4 and 1.0.7, the function font.setexpansion has been introduced (but we're not using it for now).

485 ⟨*lua-*⟩
486 ⟨*debug*⟩\MT@dinfo@nl0{this is luatex (\the\luatexversion)}

\MT@lua   Communicate with lua. Beginning with LuaTEX 0.36, \directlua no longer requires a state number.

---

2   This may have been changed earlier, but I'm no longer able to find out when (the last version that actually works for me is 0.40).

```
487 \let\MT@lua\directlua
488 \def\MT@luatex@no{6}
489 \ifnum\luatexversion<110
490   \def\MT@luatex@no{5}
491   \ifnum\luatexversion<90
492     \def\MT@luatex@no{4}
493     \ifnum\luatexversion<85
494       \def\MT@luatex@no{3}
495       \ifnum\luatexversion<62
496         \def\MT@luatex@no{2}
497         \ifnum\luatexversion<36
498           \def\MT@lua{\directlua0}
499           \def\MT@luatex@no{1}
500         \fi
501       \fi
502     \fi
503   \fi
504 \fi
505 ⟨debug⟩\MT@dinfo@nl{0}{luatex no.: \MT@luatex@no}
506 ⟨/lua-⟩
```

Abort if no capable engine found.

```
507 ⟨*package|letterspace⟩
508 \ifMT@engine@unfit
509   \MT@warning@nl{You
510     \ifx\MT@engine\relax
511       don't seem to be using pdftex%
512 ⟨package⟩        , luatex or xetex%
513 ⟨letterspace⟩       \space or luatex%
514     .\MessageBreak `\MT@MT' only works with these engines.%
515   \else
516     are using a \MT@engine tex version older than
517 ⟨package⟩        \MT@engine@minversion
518 ⟨letterspace⟩       \MT@pdf@or@lua{1.40}{0.62}%
519     .\MessageBreak `\MT@MT' does not work with this version.%
520     \MessageBreak Please install a newer version of \MT@engine tex.%
521   \fi
522   \MessageBreak I will quit now}
523   \MT@clear@options
524 \endinput\fi
525 ⟨/package|letterspace⟩
```

Still there? Then we can begin: We need the keyval package, including the 'new' \KV@@sp@def implementation. For the patch option, we use etoolbox, which requires e-TEX.

```
526 ⟨*package|letterspace⟩
527 \RequirePackage{keyval}[1997/11/10]
528 ⟨*package⟩
529 ^^X\RequirePackage{etoolbox}
```

\MT@toks   We need a token register.

```
530 \newtoks\MT@toks
```

\ifMT@if@   A scratch if.

```
531 \newif\ifMT@if@
```

1.1.4   Declarations

\ifMT@protrusion   These are the global switches . . .

\ifMT@expansion
\ifMT@auto
\ifMT@selected
\ifMT@noligatures
\ifMT@draft
\ifMT@disable
\ifMT@spacing
\ifMT@kerning
\ifMT@tracking
\ifMT@babel

```
532 \newif\ifMT@protrusion
533 \newif\ifMT@expansion
534 \newif\ifMT@auto
```

```
535 \newif\ifMT@selected
536 \newif\ifMT@noligatures
537 \newif\ifMT@draft
538 \newif\ifMT@disable
539 \newif\ifMT@spacing
540 \newif\ifMT@kerning
541 \newif\ifMT@tracking
542 \newif\ifMT@babel
```

[This line intentionally left blank.]

`\MT@pr@level` . . . and numbers.

`\MT@ex@level`
`\MT@pr@factor`
`\MT@ex@factor`
`\MT@sp@factor`
`\MT@kn@factor`

```
543 \let\MT@pr@level\tw@
544 \let\MT@ex@level\tw@
545 \let\MT@pr@factor\@m
546 \let\MT@ex@factor\@m
547 \let\MT@sp@factor\@m
548 \let\MT@kn@factor\@m
```

`\MT@pr@unit` Default unit for protrusion settings is character width, for spacing space, for kerning
`\MT@sp@unit` (and tracking) 1 em.
`\MT@kn@unit`

```
549 \let\MT@pr@unit\@empty
550 \let\MT@sp@unit\m@ne
551 \def\MT@kn@unit{1em}
```

`\MT@stretch` Expansion settings.
`\MT@shrink`
`\MT@step`

```
552 \let\MT@stretch\m@ne
553 \let\MT@shrink \m@ne
554 \let\MT@step   \m@ne
```

`\MT@pr@min` Minimum and maximum values allowed by pdfTeX.
`\MT@pr@max`
`\MT@ex@min`
`\MT@ex@max`
`\MT@sp@min`
`\MT@sp@max`
`\MT@kn@min`
`\MT@kn@max`
`\MT@tr@min`
`\MT@tr@max`

```
555 \def\MT@pr@min{-\@m}
556 \let\MT@pr@max\@m
557 \let\MT@ex@min\z@
558 \let\MT@ex@max\@m
559 \def\MT@sp@min{-\@m}
560 \let\MT@sp@max\@m
561 \def\MT@kn@min{-\@m}
562 \let\MT@kn@max\@m
563 ⟨/package⟩
564 \def\MT@tr@min{-\@m}
565 \let\MT@tr@max\@m
566 ⟨*package⟩
```

`\MT@factor@default` Default factor.

```
567 \def\MT@factor@default{1000 }
```

`\MT@stretch@default` Default values for expansion.
`\MT@shrink@default`

```
568 \def\MT@stretch@default{20 }
569 \def\MT@shrink@default{20 }
```

`\MT@letterspace` Default value for letterspacing (in thousandths of 1 em).
`\MT@letterspace@default`

```
570 ⟨/package⟩
571 \let\MT@letterspace\m@ne
572 \def\MT@letterspace@default{100}
573 ⟨*package⟩
```

`\ifMT@document` Our private test whether we're still in the preamble.

```
574 \newif\ifMT@document
575 ⟨/package⟩
576 ⟨/package|letterspace⟩
```

1.1.5 Auxiliary macros

\MT@requires@pdftex   For definitions that depend on a particular pdfTeX resp. LuaTeX version.

\MT@requires@luatex

```
577 ⟨*pdf-|lua-⟩
578 \def
579 ⟨pdf-⟩  \MT@requires@pdftex%
580 ⟨lua-⟩  \MT@requires@luatex%
581  #1{\ifnum
582 ⟨pdf-⟩  \MT@pdftex@no
583 ⟨lua-⟩  \MT@luatex@no
584     <#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}
585 ⟨lua-&debug⟩\MT@requires@luatex4{\MT@lua{tex.enableprimitives('pdf',{'tracingfonts'})}}}\relax
586 ⟨pdf-&debug⟩\MT@requires@pdftex6{
587 ⟨debug⟩\pdftracingfonts=1
588 ⟨pdf-&debug⟩}}\relax
589 ⟨/pdf-|lua-⟩
```

Some functions are loaded from a dedicated lua file. This avoids character escaping problems and incompatibilities between versions of LuaTeX. Unless running a recent LaTeX, we load the luatexbase package.

```
590 ⟨lua-⟩\@ifl@t@r\fmtversion{2016/01/01}\relax{\RequirePackage{luatexbase}}
```

We load luaotfload, because some of its functions are required in microtype.lua. This eliminates the need for the user to load fontspec before microtype. There will hardly be any LuaTeX documents that don't load this package, anyway. Since 2017/01/01, it is already loaded in the format.

```
591 ⟨lua-⟩\@ifl@t@r\fmtversion{2017/01/01}\relax{\RequirePackage{luaotfload}}
592 ⟨letterspace⟩\MT@pdf@or@lua\relax{
593 ⟨letterspace⟩\ifx\newluafunction\@undefined \input ltluatex \fi
594 ⟨lua-|letterspace⟩\MT@lua{require("microtype")}}
595 ⟨letterspace⟩}
```

Here it begins. The module was contributed by Élie Roux.

```
596 ⟨*luafile⟩
597
598 function microtype.info(...)
599   luatexbase.module_info("microtype",...)
600 end
601
602 local find      = string.find
603 local match     = string.match
604 local tex_write = tex.write
605
606 local catpackage
607 if luatexbase.registernumber then
608   catpackage = luatexbase.registernumber("catcodetable@atletter") -- LaTeX
609 else
610   catpackage = luatexbase.catcodetables.CatcodeTableLaTeXAtLetter -- luatexbase
611 end
612 function microtype.sprint (...)
613   tex.sprint(catpackage, ...)
614 end
615
```

We need the function math.tointeger, which is missing in older LuaTeX versions, and ConTeXt (inherited via luaotfload) faultily overwrites its own definition. The following is the (correct) definition from l-math.lua.

```
616 if not math.tointeger or not pcall(math.tointeger,0) then
617   math.mininteger=-0x4FFFFFFFFFFF
618   math.maxinteger=0x4FFFFFFFFFFF
619   local floor=math.floor
620   function math.tointeger(n)
```

```
621      local f=floor(n)
622      return f==n and f or nil
623    end
624 end
625
626 ⟨/luafile⟩
```

To be continued, but first back to primitives.

\MT@glet   Here's the forgotten one (finally implemented in LuaTEX).

```
627 ⟨lua-⟩\MT@requires@luatex6{\let\MT@glet\glet}\relax
628 ⟨*package|letterspace⟩
629 \def\MT@glet{\global\let}
```

\MT@exp@cs   Commands to create command sequences. Those that are going to be defined
\MT@exp@gcs   globally should be created inside a group so that the save stack won't explode.

```
630 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
631 ⟨*package⟩
632 \def\MT@exp@gcs#1#2{\begingroup\expandafter\endgroup\expandafter#1\csname#2\endcsname}
```

\MT@def@n   This is \@namedef and global.
\MT@gdef@n

```
633 \def\MT@def@n{\MT@exp@cs\def}
634 \def\MT@gdef@n{\MT@exp@gcs\gdef}
```

\MT@edef@n   Its expanding versions.
\MT@xdef@n

```
635 ⟨/package⟩
636 \def\MT@edef@n{\MT@exp@cs\edef}
637 ⟨*package⟩
638 \def\MT@xdef@n{\MT@exp@gcs\xdef}
```

\MT@let@nc   \let a \csname sequence to a command.
\MT@glet@nc

```
639 \def\MT@let@nc{\MT@exp@cs\let}
640 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}
```

\MT@let@cn   \let a command to a \csname sequence.

```
641 ⟨/package⟩
642 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
643 ⟨*package⟩
```

\MT@let@nn   \let a \csname sequence to a \csname sequence.
\MT@glet@nn

```
644 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
645 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}
```

\MT@@font   Remove trailing space from the font name.

```
646 \def\MT@@font{\expandafter\string\MT@font}
```

\MT@exp@one@n   Expand the second token once and enclose it in braces.

```
647 ⟨/package⟩
648 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}
```

\MT@exp@two@c   Expand the next two tokens after ⟨#1⟩ once.

```
649 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
650 ⟨*package⟩
```

\MT@exp@two@n   Expand the next two tokens after ⟨#1⟩ once and enclose them in braces.

```
651 \def\MT@exp@two@n#1#2#3{%
652    \expandafter\expandafter\expandafter
653      #1\expandafter\expandafter\expandafter
654        {\expandafter#2\expandafter}\expandafter{#3}}
```

You do not wonder why \MT@exp@one@c doesn't exist, do you?

\MT@ifdefined@c@T   Wrapper for testing whether command resp. \csname sequence is defined. If we
\MT@ifdefined@c@TF   are running e-TEX, we will use its primitives \ifdefined and \ifcsname, which
\MT@ifdefined@n@T
\MT@ifdefined@n@TF

decreases memory use substantially.

```
655 \def\MT@ifdefined@c@T#1{%
656 ^^X  \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
657 ^^Q  \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
658 }
659 ⟨/package⟩
660 \def\MT@ifdefined@c@TF#1{%
661 ^^X  \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
662 ⟨package⟩^^Q  \ifx#1\@undefined
663 ⟨package⟩^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
664 }
665 \def\MT@ifdefined@n@T#1{%
666 ^^X  \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
667 ⟨package⟩^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
668 ⟨package⟩^^Q    \expandafter\@gobble\else\expandafter\@firstofone\fi
669 }
670 \def\MT@ifdefined@n@TF#1{%
671 ^^X  \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
672 ⟨package⟩^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
673 ⟨package⟩^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
674 }
675 ⟨*package⟩
```

\MT@detokenize@n
\MT@detokenize@c
\MT@rem@last@space

Translate a macro into a token list. With e-TeX, we can use \detokenize. We also need to remove the last trailing space; and only the last one – therefore the fiddling (and the \string isn't perfect, of course).

```
676 \def\MT@detokenize@n#1{%
677 ^^X  \expandafter\MT@rem@last@space\detokenize{#1} \@nil
678 ^^Q  \string#1%
679 }
680 \def\MT@detokenize@c#1{%
681 ^^X  \MT@exp@one@n\MT@detokenize@n#1%
682 ^^Q  \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
683 }
684 \def\MT@rem@last@space#1 #2{#1%
685   \ifx\@nil#2\else \space
686   \expandafter\MT@rem@last@space\expandafter#2\fi
687 }
```

\MT@ifempty

Test whether argument is empty.

```
688 ⟨/package⟩
689 \begingroup
690 \catcode`\%=12
691 \catcode`\&=14
692 \gdef\MT@ifempty#1{&
693   \if %#1%&
694     \expandafter\@firstoftwo
695   \else
696     \expandafter\@secondoftwo
697   \fi
698 }
699 \endgroup
700 ⟨*package⟩
```

\MT@ifint

Test whether argument is an integer, using an old trick by Mr. Arseneau, or the latest and greatest from pdfTeX or LuaTeX (which also allows negative numbers, as required by the letterspace option).

```
701 ⟨/package⟩
702 ⟨/package|letterspace⟩
703 ⟨pdf-⟩\MT@requires@pdftex6{
704 ⟨letterspace⟩\MT@pdf@or@lua{
705 ⟨*pdf-|letterspace⟩
706 \def\MT@ifint#1{%
707   \ifcase\pdfmatch{^-*[0-9]+ *$}{#1}\relax
```

```
708      \expandafter\@secondoftwo
709    \else
710      \expandafter\@firstoftwo
711    \fi
712 }
713 }{
```
714 ⟨/pdf-|letterspace⟩
715 ⟨*pdf-|xe-|letterspace⟩
```
716 \def\MT@ifint#1{%
717    \if!\ifnum9<1#1!\else?\fi
718      \expandafter\@firstoftwo
719    \else
720      \expandafter\@secondoftwo
721    \fi
722 }
```
723 ⟨/pdf-|xe-|letterspace⟩
724 ⟨pdf-|letterspace⟩}
725 ⟨lua-⟩\def\MT@ifint#1{\csname\MT@lua{microtype.if_int([[#1]])}\endcsname}
726 ⟨*luafile⟩
```
727 local function if_int(s)
728    if find(s,"^-*[0-9]+ *$") then
729      tex_write("@firstoftwo")
730    else
731      tex_write("@secondoftwo")
732    end
733 end
734 microtype.if_int = if_int
735
```
736 ⟨/luafile⟩

\MT@ifdimen   Test whether argument is dimension (or number). (nd and nc are new Didot resp.
Cicero, added in pdfTEX 1.30; px is a pixel.)

737 ⟨*pdf-⟩
```
738 \MT@requires@pdftex6{
739 \def\MT@ifdimen#1{%
740    \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
741                    (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
742      \expandafter\@secondoftwo
743    \else
744      \expandafter\@firstoftwo
745    \fi
746 }
747 }{
```
748 ⟨/pdf-⟩
749 ⟨*pdf-|xe-⟩
```
750 \def\MT@ifdimen#1{%
751    \setbox\z@=\hbox{%
752      \MT@count=1#1\relax
753      \ifnum\MT@count=\@ne
754        \aftergroup\@secondoftwo
755      \else
756        \aftergroup\@firstoftwo
757      \fi
758    }%
759 }
```
760 ⟨/pdf-|xe-⟩
761 ⟨pdf-⟩}
762 ⟨lua-⟩\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen([[#1]])}\endcsname}
763 ⟨*luafile⟩
```
764 local function if_dimen(s)
765    if (find(s, "^-*[0-9]+(%a*) *$") or
766        find(s, "^-*[0-9]*[.,][0-9]+(%a*) *$")) then
767      tex_write("@firstoftwo")
768    else
769      tex_write("@secondoftwo")
```

```
770     end
771   end
772   microtype.if_dimen = if_dimen
773
774   ⟨/luafile⟩
```

\MT@ifdim   Compare floating point numbers.

```
775   ⟨*package⟩
776   \def\MT@ifdim#1#2#3{%
777     \ifdim #1\p@ #2 #3\p@
778       \expandafter\@firstoftwo
779     \else
780       \expandafter\@secondoftwo
781     \fi
782   }
783   ⟨/package⟩
```

\MT@ifstreq   Test whether two strings (fully expanded) are equal.

```
784   ⟨*pdf-|xe-⟩
785   ⟨pdf-⟩\MT@requires@pdftex5{
786   \def\MT@ifstreq#1#2{%
787   ⟨pdf-⟩  \ifnum\pdfstrcmp{#1}{#2}=\z@
788   ⟨xe-⟩   \ifnum\strcmp{#1}{#2}=\z@
789       \expandafter\@firstoftwo
790     \else
791       \expandafter\@secondoftwo
792     \fi
793   }
794   ⟨/pdf-|xe-⟩
795   ⟨*pdf-⟩
796   }{
797   \def\MT@ifstreq#1#2{%
798     \edef\MT@res@a{#1}%
799     \edef\MT@res@b{#2}%
800     \ifx\MT@res@a\MT@res@b
801       \expandafter\@firstoftwo
802     \else
803       \expandafter\@secondoftwo
804     \fi
805   }
806   }
807   ⟨/pdf-⟩
808   ⟨lua-⟩\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq([[#1]],[[#2]])}\endcsname}
809   ⟨*luafile⟩
810   local function if_str_eq(s1, s2)
811     if s1 == s2 then
812       tex_write("@firstoftwo")
813     else
814       tex_write("@secondoftwo")
815     end
816   end
817   microtype.if_str_eq = if_str_eq
818
819   ⟨/luafile⟩
```

\MT@xadd   Add item to a list.

```
820   ⟨*package⟩
821   \def\MT@xadd#1#2{%
822     \ifx#1\relax
823       \xdef#1{#2}%
824     \else
825       \xdef#1{#1#2}%
826     \fi
827   }
```

\MT@xaddb   Add item to the beginning.

```
828 \def\MT@xaddb#1#2{%
829   \ifx#1\relax
830     \xdef#1{#2}%
831   \else
832     \xdef#1{#2#1}%
833   \fi
834 }
835 ⟨/package⟩
```

\MT@map@clist@n   Run ⟨#2⟩ on all elements of the comma list ⟨#1⟩. This and the following is modelled
\MT@map@clist@c   after LaTeX3 commands.
\MT@map@clist@
\MT@clist@function
\MT@clist@break

```
836 ⟨*package|letterspace⟩
837 \def\MT@map@clist@n#1#2{%
838   \ifx\@empty#1\else
839     \def\MT@clist@function##1{#2}%
840     \MT@map@clist@#1,\@nil,\@nnil
841   \fi
842 }

843 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
844 \def\MT@map@clist@#1,{%
845   \ifx\@nil#1%
846     \expandafter\MT@clist@break
847   \fi
848   \MT@clist@function{#1}%
849   \MT@map@clist@
850 }
851 \let\MT@clist@function\@gobble
852 \def\MT@clist@break#1\@nnil{}
853 ⟨*package⟩
```

\MT@map@tlist@n   Execute ⟨#2⟩ on all elements of the token list ⟨#1⟩. \MT@tlist@break can be used
\MT@map@tlist@c   to jump out of the loop.
\MT@map@tlist@
\MT@tlist@break

```
854 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
855 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
856 \def\MT@map@tlist@#1#2{%
857   \ifx\@nnil#2\else
858     #1{#2}%
859     \expandafter\MT@map@tlist@
860     \expandafter#1%
861   \fi
862 }
863 \def\MT@tlist@break#1\@nnil{\fi}
```

\ifMT@inlist@   Test whether item ⟨#1⟩ is in comma list ⟨#2⟩. Using \pdfmatch would be slower.
\MT@in@clist

```
864 \newif\ifMT@inlist@
865 \def\MT@in@clist#1#2{%
866   \def\MT@res@a##1,#1,##2##3\@nnil{%
867     \ifx##2\@empty
868       \MT@inlist@false
869     \else
870       \MT@inlist@true
871     \fi
872   }%
873   \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
874 }
```

\MT@rem@from@clist Remove item ⟨#1⟩ from comma list ⟨#2⟩. This is basically \@removeelement from
                   ltcntrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!

```
875 \def\MT@rem@from@clist#1#2{%
876   \def\MT@res@a##1,#1,##2{\MT@res@a{##1,##2}\MT@res@b}%
877   \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
878   \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
```

```
879 }
```

\MT@in@tlist
\MT@in@tlist@

Test whether item is in token list. Since this isn't too elegant, I thought that at least here, \pdfmatch would be more efficient – however, it turned out to be even slower than this solution.

```
880 \def\MT@in@tlist#1#2{%
881   \MT@inlist@false
882   \def\MT@res@a{#1}%
883   \MT@map@tlist@c#2\MT@in@tlist@
884 }
885 \def\MT@in@tlist@#1{%
886   \edef\MT@res@b{#1}%
887   \ifx\MT@res@a\MT@res@b
888     \MT@inlist@true
889     \expandafter\MT@tlist@break
890   \fi
891 }
```

\MT@in@rlist
\MT@in@rlist@
\MT@in@rlist@@
\MT@size@name

Test whether size \MT@size is in a list of ranges. Store the name of the list in \MT@size@name

```
892 \def\MT@in@rlist#1{%
893   \MT@inlist@false
894   \MT@map@tlist@c#1\MT@in@rlist@
895 }
896 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
897 \def\MT@in@rlist@@#1#2#3{%
898   \MT@ifdim{#2}=\m@ne{%
899     \MT@ifdim{#1}=\MT@size
900       \MT@inlist@true
901       \relax
902   }{%
903     \MT@ifdim\MT@size<{#1}\relax{%
904       \MT@ifdim\MT@size<{#2}%
905         \MT@inlist@true
906         \relax
907     }%
908   }%
909   \ifMT@inlist@
910     \def\MT@size@name{#3}%
911     \expandafter\MT@tlist@break
912   \fi
913 }
```

\MT@loop
\MT@iterate
\MT@repeat

This is the same as LaTeX's \loop, which we mustn't use, since this could confuse an outer \loop in the document.

```
914 ⟨/package⟩
915 \def\MT@loop#1\MT@repeat{%
916   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
917   \MT@iterate \let\MT@iterate\relax
918 }
919 \let\MT@repeat\fi
```

\MT@while@num

Execute ⟨#3⟩ from ⟨#1⟩ up to (excluding) ⟨#2⟩ (much faster than LaTeX's \@whilenum).

```
920 \def\MT@while@num#1#2#3{%
921   \@tempcnta#1\relax
922   \MT@loop #3%
923     \advance\@tempcnta \@ne
924     \ifnum\@tempcnta < #2\MT@repeat
925 }
926 ⟨/package|letterspace⟩
```

\MT@if@luaotf@font

For fonts loaded by luaotfload we query the font's table.

```
927 ⟨letterspace⟩\MT@pdf@or@lua{\let\MT@if@luaotf@font\@secondoftwo}{
928 ⟨*lua-|letterspace⟩
```

```
929 \def\MT@if@luaotf@font{\csname\MT@lua{%
930   microtype.if_luaotf_font()
931   }\endcsname
932 }
```
933 ⟨/lua-|letterspace⟩
934 ⟨letterspace⟩}
935 ⟨*luafile⟩
```
936 local function if_luaotf_font()
937   local thefont = font.getfont(font.current())
938   if thefont and ( thefont.format == "opentype" or thefont.format == "truetype" )
939     then tex.write("@firstoftwo")
940     else tex.write("@secondoftwo")
941   end
942 end
943 microtype.if_luaotf_font = if_luaotf_font
944
```
945 ⟨/luafile⟩

\MT@do@font   Execute ⟨#1⟩ 256 times,

946 ⟨pdf-|letterspace⟩\def\MT@do@font{\MT@while@num\z@\@cclvi}

resp. for the whole font for LuaTEX, if it's a Unicode font.

947 ⟨*lua-⟩
```
948 \def\MT@do@font#1{%
949   \MT@if@luaotf@font{%
950     \def\MT@dofont@function{#1}%
951     \MT@lua{microtype.do_font()}%
952   }{\MT@while@num\z@\@cclvi{#1}}%
953 }
```
954 ⟨/lua-⟩

This is the lua function, which is much faster than looping through all glyphs in TEX. Legacy fonts (which this function should never work on) don't contain a v.index field. Our test whether i is larger than 1114111 may seem strange, but with the HarfBuzz renderer, we are not guaranteed to get a number within the Unicode range.

955 ⟨*luafile⟩
```
956 local function do_font()
957   local thefont = font.getfont(font.current())
958   if thefont then
959     for i,v in next,thefont.characters do
960       if v.index == nil or ( v.index > 0 and i < 1114112 ) then
961         microtype.sprint([[\@tempcnta=]]..i..[[\relax\MT@dofont@function]])
962       end
963     end
964   end
965 end
966 microtype.do_font = do_font
967
```
968 ⟨/luafile⟩

The X∃TEX variant (it's slow … !).

969 ⟨*xe-⟩
```
970 \def\MT@do@font#1{%
971   \@tempcnta=\z@
972   \MT@loop
973     \iffontchar\MT@font\@tempcnta #1\fi
974     \advance\@tempcnta\@ne
975     \ifnum\@tempcnta < \XeTeXlastfontchar\MT@font \MT@repeat
976 }
```
977 ⟨/xe-⟩
978 ⟨*package⟩

\MT@count   Increment macro ⟨#1⟩ by one. Saves using up too many counters. The e-TEX way is
\MT@increment

slightly faster.

```
979 \newcount\MT@count
980 \def\MT@increment#1{%
981 ^^X  \edef#1{\number\numexpr #1 + 1\relax}%
982 ^^Q  \MT@count=#1\relax
983 ^^Q  \advance\MT@count \@ne
984 ^^Q  \edef#1{\number\MT@count}%
985 }
```

\MT@scale   Multiply and divide a counter. If we are using e-TEX, we will use its \numexpr
primitive. This has the advantage that it is less likely to run into arithmetic overflow.
The result of the division will be rounded instead of truncated. Therefore, we'll get
a different (more accurate) result in about half of the cases.

```
986 \def\MT@scale#1#2#3{%
987 ^^Q  \multiply #1 #2\relax
988   \ifnum #3 = \z@
989 ^^X    #1=\numexpr #1 * #2\relax
990   \else
991 ^^X    #1=\numexpr #1 * #2 / #3\relax
992 ^^Q    \divide #1 #3\relax
993   \fi
994 }
```

\MT@abbr@pr   Some abbreviations. Thus, we can have short command names but full-length log
\MT@abbr@ex   output.

```
\MT@abbr@pr@c    995 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c    996 \def\MT@abbr@ex{expansion}
                 997 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@pr@inh  998 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@ex@inh  999 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@nl     1000 \def\MT@abbr@ex@inh{expansion inheritance}
               1001 \def\MT@abbr@nl{noligatures}
\MT@abbr@sp    1002 \def\MT@abbr@sp{spacing}
\MT@abbr@sp@c  1003 \def\MT@abbr@sp@c{interword spacing codes}
               1004 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@sp@inh 1005 \def\MT@abbr@kn{kerning}
\MT@abbr@kn    1006 \def\MT@abbr@kn@c{kerning codes}
\MT@abbr@kn@c  1007 \def\MT@abbr@kn@inh{kerning inheritance}
               1008 \def\MT@abbr@tr{tracking}
\MT@abbr@kn@inh 1009 \def\MT@abbr@tr@c{tracking amount}
\MT@abbr@tr
\MT@rbba@protrusion
\MT@abbr@tr@c
```

These we also need the other way round.

```
\MT@rbba@expansion 1010 \def\MT@rbba@protrusion{pr}
\MT@rbba@spacing   1011 \def\MT@rbba@expansion{ex}
                   1012 \def\MT@rbba@spacing{sp}
\MT@rbba@kerning   1013 \def\MT@rbba@kerning{kn}
\MT@rbba@tracking  1014 \def\MT@rbba@tracking{tr}
```

\MT@features   We can work on these lists to save some guards in the dtx file.

```
\MT@features@long 1015 \def\MT@features{pr,ex,sp,kn,tr}
                  1016 \def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}
```

\MT@is@feature   Whenever an optional argument accepts a list of features, we can use this com-
mand to check whether a feature exists in order to prevent a rather confusing
'Missing \endcsname inserted' error message. The feature (long form) must be in
⟨#1⟩, the type of list to ignore in ⟨#2⟩, then comes the action.

```
1017 \def\MT@is@feature#1#2{%
1018   \MT@in@clist{#1}\MT@features@long
1019   \ifMT@inlist@
1020     \expandafter\@firstofone
1021   \else
1022     \MT@error{`#1' is not an available micro-typographic\MessageBreak
```

```
1023        feature. Ignoring #2}{Available features are: `\MT@features@long'.}%
1024      \expandafter\@gobble
1025    \fi
1026 }
```

1.1.6 Compatibility

For the record, the following LaTeX kernel commands will be modified by `microtype`:

- \pickup@font

- \do@subst@correction

- \add@accent (all in section 1.2.9)

- \showhyphens (in section 1.4.6)

The `wordcount` package redefines the font-switching commands, which will break `microtype`. Since `microtype` doesn't have an effect on the number of words in the document anyway, we will simply disable ourselves.

```
1027 \@ifl@aded{tex}{wordcount}{%
1028    \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
1029      Disabling `\MT@MT', since it wouldn't work}%
1030    \MT@clear@options\endinput}\relax
```

The `minimal` class doesn't define any size commands other than \normalsize, which will result in lots of warnings. Therefore we issue a warning about the warnings.

```
1031 \@ifclassloaded{minimal}{%
1032    \MT@warning@nl{Detected the `minimal' class.\MessageBreak
1033      Expect lots of warnings and some malfunctions.\MessageBreak
1034      You might want to use a proper class instead}%
1035 }\relax
```

\MT@setup@ The setup is deferred until the end of the preamble. This has a couple of advantages: \microtypesetup can be used to change options later on in the preamble, and fonts don't have to be set up before `microtype`.

```
1036 ⟨/package⟩
1037 ⟨*package|letterspace⟩
1038 ⟨plain⟩\MT@requires@latex1{
1039 \let\MT@setup@\@empty
```

\MT@addto@setup We use our private hook to have better control over the timing. This will also work with `eplain`, but not with `miniltx` alone.

```
1040 \def\MT@addto@setup{\g@addto@macro\MT@setup@}
```

Don't hesitate with `miniltx`.

```
1041 ⟨plain⟩}{\let\MT@addto@setup\@firstofone}
```

\MT@with@package@T We almost never do anything if a package is not loaded.

```
1042 \def\MT@with@package@T#1{\@ifpackageloaded{#1}\@firstofone\@gobble}
1043 ⟨/package|letterspace⟩
1044 ⟨*package⟩
```

\MT@with@babel@and@T LaTeX's \@ifpackagewith ignores the class options.

```
1045 \def\MT@with@babel@and@T#1{%
1046    \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
1047      \@expandtwoargs\MT@in@clist{#1}
1048        {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
1049      \ifMT@inlist@\expandafter\@secondoftwo\else\expandafter\@firstofone\fi
1050    }\@gobble
1051 }
```

`\MT@ledmac@setup`   The `ledmac` package first saves each paragraph in a box, from which it then splits off the lines one by one. This will destroy character protrusion. (There aren't any problems with the `lineno` package, since it takes a different approach.) — ... — After much to and fro, the situation has finally settled and there is a fix. Beginning with pdfTeX version 1.21b together with `ledpatch.sty` as of 2005/06/02 (v0.4), character protrusion will work at last.

Peter Wilson was so kind to provide the `\l@dunhbox@line` hook in `ledmac` to allow for protrusion. `\leftmarginkern` and `\rightmarginkern` are new primitives of pdfTeX 1.21b (aka. 1.30.0). They are also part of recent XƎTEX. The successor packages `eledmac` and `reledmac` are also supported.

```
1052 ⟨/package⟩
1053 ⟨pdf-⟩\MT@requires@pdftex5{
1054 ⟨*pdf-|lua-|xe-⟩
1055   \def\MT@ledmac@setup{%
1056     \ifMT@protrusion
1057       \MT@ifdefined@c@TF\l@dunhbox@line{%
```

`\MT@led@unhbox@line`   Hook.

```
1058       \MT@info@nl{Patching ((r)e)ledmac to enable character protrusion}%
1059       \let\MT@led@unhbox@line\l@dunhbox@line
1060       \renewcommand*{\l@dunhbox@line}[1]{%
1061         \ifhbox##1%
1062           \kern\leftmarginkern##1%
1063           \expandafter\MT@led@unhbox@line\expandafter##1\expandafter
1064           \kern\rightmarginkern##1%
1065         \fi
1066       }%
1067     }{%
1068       \MT@warning@nl{%
1069         Character protrusion in paragraphs with line\MessageBreak
1070         numbering will only work if you update ledmac,\MessageBreak
1071         or use one of its successors, eledmac or reledmac}%
1072     }%
1073   \fi
1074 }
1075 ⟨/pdf-|lua-|xe-⟩
1076 ⟨*pdf-⟩
1077 }{
1078   \def\MT@ledmac@setup{%
1079     \ifMT@protrusion
1080       \MT@warning@nl{%
1081         The pdftex version you are using does not allow\MessageBreak
1082         character protrusion in paragraphs with line\MessageBreak
1083         numbering by the `((r)e)ledmac' package.\MessageBreak
1084         Upgrade pdftex to version 1.30 or later}%
1085     \fi
1086   }
1087 }
1088 ⟨/pdf-⟩
```

The `shapepar` package (v2.2) fixes this in a similar manner by itself, so we don't have to bother.

`\MT@restore@p@h`   Restore meaning of `\%` and `\#`.

```
1089 ⟨*package|letterspace⟩
1090 ⟨*package⟩
1091 \def\MT@restore@p@h{\chardef\%`\% \chardef\#`\# }
```

`\ifMT@fontspec`   Two new conditionals for use with XƎTEX or LuaTEX.

`\ifMT@xunicode`
```
1092 \newif\ifMT@fontspec
1093 \MT@with@package@T{fontspec}\MT@fontspectrue
```

```
1094 \newif\ifMT@xunicode
1095 \MT@with@package@T{xunicode}\MT@xunicodetrue
```

We need the correct value of the former for configuration commands inside the preamble (to get the default families right).

```
1096 \@ifl@t@r\fmtversion{2020/10/01}
1097   {\IfFormatAtLeastTF{2021/11/15}
1098     {\AddToHook{package/fontspec/after}{\MT@fontspectrue}}
1099     {\AddToHook{package/after/fontspec}{\MT@fontspectrue}}}\relax
```

\MT@maybe@gobble@with@tikz
\MT@tikz@setup

If \tikz@expandcount is greater than zero, we're inside or at the end of a tikz node, where we don't want to adjust spacing after letterspacing, lest we disturb tikz. This is used in \MT@afteraftergroup, and we don't need it for letterspace.

```
1100 \let\MT@maybe@gobble@with@tikz\@firstofone
1101 \def\MT@tikz@setup{%
1102   \def\MT@maybe@gobble@with@tikz{%
1103     \ifnum\tikz@expandcount>\z@
1104       \expandafter\@gobble
1105     \else
1106       \expandafter\@firstofone
1107     \fi}}
```

\MT@setupfont@hook

This hook will be executed every time a font is set up (inside a group).

In the preamble, we check for the packages each time a font is set up. Thus, it will work regardless when the packages are loaded.

Even for packages that don't activate any characters in the preamble (like babel and csquotes), we have to check here, too, in case they were loaded before microtype, and a font is loaded \AtBeginDocument, before microtype. (This is no longer needed, since the complete setup is now deferred until the end of the preamble. However, it is still necessary for defersetup=false.)

```
1108 \def\MT@setupfont@hook{%
```

Spanish (as well as Galician and Mexican) babel modify \%, storing the original meaning in \percentsign.

```
1109   \MT@if@false
1110   \MT@with@babel@and@T{spanish} \MT@if@true
1111   \MT@with@babel@and@T{galician}\MT@if@true
1112   \MT@with@babel@and@T{mexican} \MT@if@true
1113   \ifMT@if@\MT@ifdefined@c@T\percentsign{\let\%\percentsign}\fi
```

Using \@disablequotes, we can restore the original meaning of all characters made active by csquotes. (It would be doable for older versions, too, but we won't bother.)

```
1114   \MT@with@package@T{csquotes}{%
1115     \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax}%
```

hyperref redefines \% and \# inside a \url. We restore the original meanings (which we can only hope are correct). Same for tex4ht and mathastext.

```
1116   \MT@if@false
1117   \MT@with@package@T{hyperref}   \MT@if@true
1118   \MT@with@package@T{tex4ht}      \MT@if@true
1119   \MT@with@package@T{mathastext}\MT@if@true
1120   \ifMT@if@\MT@restore@p@h\fi
1121   \MT@with@package@T{tikz}\MT@tikz@setup
1122 }
```

Check again at the end of the preamble.

```
1123 ⟨/package⟩
1124 \MT@addto@setup{%
1125 ⟨*package⟩
```

Our competitor, the `pdfcprot` package, must not be tolerated!

```
1126  \MT@with@package@T{pdfcprot}{%
1127    \MT@error{Detected the `pdfcprot' package!\MessageBreak
1128            `\MT@MT' and `pdfcprot' may not be used together}{%
1129 The `pdfcprot' package provides an interface to character protrusion.\MessageBreak
1130 So does the `\MT@MT' package. Using both packages at the same\MessageBreak
1131 time will almost certainly lead to undesired results. Have your choice!}%
1132    }%
1133  \MT@with@package@T  {ledmac}\MT@ledmac@setup
1134  \MT@with@package@T {eledmac}\MT@ledmac@setup
1135  \MT@with@package@T{reledmac}\MT@ledmac@setup
1136  \MT@with@package@T{xunicode}\MT@xunicodetrue
1137  \MT@with@package@T{fontspec}\MT@fontspectrue
```

We can clean up `\MT@setupfont@hook` now.

```
1138  \MT@glet\MT@setupfont@hook\@empty
```

`microtype` is so so loquacious … Sometimes you just want to silence it when debugging a document.

```
1139  %\gdef\MT@setupfont@hook{\tracingnone
1140  %  \MT@info{Silently doing my `magic' (Mittelbach) for font\MessageBreak\MT@@font}}%
1141  \MT@if@false
1142  \MT@with@babel@and@T{spanish} \MT@if@true
1143  \MT@with@babel@and@T{galician}\MT@if@true
1144  \MT@with@babel@and@T{mexican} \MT@if@true
1145  \ifMT@if@
1146    \g@addto@macro\MT@setupfont@hook{%
1147      \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
1148  \fi
1149  \MT@with@package@T{csquotes}{%
1150    \@ifpackagelater{csquotes}{2005/05/11}{%
1151      \g@addto@macro\MT@setupfont@hook\@disablequotes
1152      \g@addto@macro\MT@prot@hook\@disablequotes
1153    }{%
1154      \MT@warning@nl{%
1155        Should you receive warnings about unknown slot\MessageBreak
1156        numbers, try upgrading the `csquotes' package}%
1157    }%
1158  }%
```

We disable `microtype`'s additions inside `hyperref`'s `\pdfstringdef`, which redefines lots of commands. `hyperref` doesn't work with plain TEX, so in that case we don't bother.

```
1159    \MT@if@false
1160 ⟨/package⟩
1161 ⟨plain⟩  \MT@requires@latex2{
1162  \MT@with@package@T{hyperref}{%
1163    \pdfstringdefDisableCommands{%
1164 ⟨*package⟩
1165      \MT@ltx@pickupfont
1166      \let\textmicrotypecontext\@secondoftwo
1167      \let\microtypecontext\@gobble
1168 ⟨/package⟩
1169      \def\lsstyle{\pdfstringdefWarn\lsstyle}%
1170      \def\textls#1#{\pdfstringdefWarn\textls}%
1171    }%
1172 ⟨package⟩    \MT@if@true
1173  }%
1174 ⟨plain⟩  }\relax
1175 ⟨*package⟩
1176  \MT@with@package@T{tex4ht}{%
1177    \def\MT@apply@patch#1{\MT@info{Not applying patch `#1' (for tex4ht)}}%
1178    \def\MT@undo@patch#1{\MT@info{Not undoing patch `#1' (for tex4ht)}}%
1179    \MT@if@true
```

```
1180    }%
1181    \MT@with@package@T{mathastext}\MT@if@true
1182    \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
```

The `listings` package makes numbers and letters active,

```
1183    \MT@with@package@T{listings}{%
1184      \g@addto@macro\MT@cfg@catcodes{%
1185        \MT@while@num{"30}{"3A}{\catcode\@tempcnta=12\relax}%
1186        \MT@while@num{"41}{"5B}{\catcode\@tempcnta=11\relax}%
1187        \MT@while@num{"61}{"7B}{\catcode\@tempcnta=11\relax}%
1188      }%
```

... and the backslash (which would lead to problems in `\MT@get@slot`).

```
1189      \g@addto@macro\MT@setupfont@hook{%
1190        \catcode`\\=\z@
```

Inside a listing, `\space` is redefined.

```
1191        \def\space{ }%
```

When loaded with the `extendedchar` option, `listings` will also redefine 8-bit active characters (`inputenc`). Luckily, this simple redefinition will make them expand to their original definition, so that they could be used in the configuration.

```
1192        \let\lst@ProcessLetter\@empty
1193      }%
1194    }%
```

Of course, using both `soul`'s and `microtype`'s letterspacing mechanisms at the same time doesn't make much sense. But `soul` can do more, e.g., underlining. The optional argument to `\textls` may not be used. Also, we have to disable expansion within `soul`'s trial run. Under plain TEX, `soul` doesn't register itself the LATEX way, so we just test for its main command.

```
1195    ⟨/package⟩
1196    \ifx\SOUL@\@undefined\else
1197      \soulregister\lsstyle 0%
1198      \soulregister\textls  1%
1199      \ifx\XeTeXrevision\@undefined
1200        \let\MT@SOUL@doword\SOUL@doword
1201        \def\SOUL@doword{\pdfadjustspacing=\z@ \MT@SOUL@doword}%
1202      \fi
1203    \fi
1204    ⟨*package⟩
1205    \MT@with@package@T{tikz}\MT@tikz@setup
```

Compatibility with the `pinyin` package (from CJK): disable `microtype` in `\py@macron`, which loads a different font for the accent. In older versions of `pinyin` (pre-4.6.0), `\py@macron` had only one argument.

```
1206    \MT@with@package@T{pinyin}{%
1207      \let\MT@orig@py@macron\py@macron
1208      \@ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
1209        \def\py@macron#1#2{%
1210          \MT@ltx@pickupfont
1211          \MT@orig@py@macron{#1}{#2}%
1212          \MT@MT@pickupfont}%
1213      }{%
1214        \def\py@macron#1{%
1215          \MT@ltx@pickupfont
1216          \MT@orig@py@macron{#1}%
1217          \MT@MT@pickupfont}%
1218      }%
1219    }%
```

The `luainputenc` package makes all characters active, which can lead into problems when the `unicode-math` package is loaded, as the latter doesn't always define

characters in LICR-conforming ways. By disabling the following command, we prevent errors; warnings about unknown slots, however, may still occur – but that's one of the unavoidable downsides of using luainputenc.

```
1220  \MT@with@package@T{unicode-math}{%
1221    \MT@let@nc{__um_sub_or_super:n}\relax
1222  }%
1223  ⟨/package⟩
1224  }
1225  ⟨*package⟩
```

1.1.7 Protrusion patches

\ifMT@patch@ok   We have to patch some macros to get protrusion right.

\MT@patch@info
\MT@patch@warn
\MT@patch@undef
\MT@patch@info@undo
```
1226  \newif\ifMT@patch@ok
1227  \def\MT@patch@info#1{\MT@info{Applying patch `#1'}}
1228  \def\MT@patch@warn#1{\MT@warning{Unable to apply patch `#1'}}
1229  \def\MT@patch@undef#1{\MT@warning{Patch `#1' undefined. Cannot apply it}}
1230  \def\MT@patch@info@undo#1{\MT@info{Reverting patch `#1'}}
```

\MT@patches@def
\MT@define@patch
Define a patch and add it to the list of patches. The third argument may contain more revert commands, but will mostly be empty.

```
1231  \let\MT@patches@def\@gobble
1232  \def\MT@define@patch#1#2#3{%
1233    \g@addto@macro\MT@patches@def{,#1}%
1234    \MT@def@n{MT@patch@@#1}{#2}%
1235    \MT@def@n{MT@patch@undo@@#1}{#3}%
1236  }
```

\MT@redefined@patches
\MT@redefine@patch
We also provide an easier way of redefining patches, which would otherwise be a bit tricky because of the timing (patches are defined *and* executed ABD).

```
1237  \let\MT@redefined@patches\@empty
1238  \def\MT@redefine@patch#1#2#3{%
1239    \g@addto@macro\MT@redefined@patches{%
1240      \MT@def@n{MT@patch@@#1}{#2}%
1241      \MT@def@n{MT@patch@undo@@#1}{#3}%
1242    }%
1243  }
```

Both macros are only allowed in the preamble.

```
1244  \@onlypreamble\MT@define@patch
1245  \@onlypreamble\MT@redefine@patch
```

\MT@append@patch
\MT@patch@patch
Wrappers around etoolbox commands. We also remember the original command to allow unpatching.

```
1246  \def\MT@append@patch#1#2{%
1247    \MT@remember@patch{#1}%
1248    \apptocmd#1{#2}\relax\MT@patch@okfalse
1249  }
1250  \def\MT@patch@patch#1#2#3{%
1251    \MT@remember@patch{#1}%
1252    \patchcmd#1{#2}{#3}\relax\MT@patch@okfalse
1253  }
```

\MT@remember@patch
Remember the original definition and add to undo command.

```
1254  \def\MT@remember@patch#1{%
1255    \MT@ifdefined@n@TF{MT@patch@saved@\string#1}\relax
1256      {\MT@let@nc{MT@patch@saved@\string#1}#1%
1257        \MT@exp@cs\g@addto@macro{MT@patch@undo@@\MT@patch@name}%
1258          {\MT@let@cn#1{MT@patch@saved@\string#1}}}%
1259  }
```

\MT@patches@applied
\MT@apply@patch

Apply a previously defined patch. With some packages, we have to reset catcodes (e.g., for the 'item' patch with Spanish babel, which makes '>' active).

```
1260 \let\MT@patches@applied\@gobble
1261 \def\MT@apply@patch#1{%
1262   \MT@patch@oktrue
1263   \MT@ifdefined@n@TF{MT@patch@@#1}
1264     {\MT@in@clist{#1}\MT@patches@applied
1265      \ifMT@inlist@
1266        \MT@warning{Patch `#1' has already been applied,\MessageBreak
1267                    cannot reapply it}%
1268      \else
1269        \let\MT@restore@catcodes\@empty
1270        \MT@with@babel@and@T{spanish} {\MT@fix@catcode{62}{12}}% >
1271        \MT@with@babel@and@T{galician}{\MT@fix@catcode{62}{12}}% >
1272        \def\MT@patch@name{#1}%
1273        \g@addto@macro\MT@patches@applied{,#1}%
1274        \@nameuse{MT@patch@@#1}%
1275        \@nameuse{MT@patch@\ifMT@patch@ok info\else warn\fi}{#1}%
1276        \MT@restore@catcodes
1277      \fi}
1278     {\MT@patch@undef{#1}}%
1279 }
```

\MT@undo@patch   Undo a patch (if indeed previously applied).

```
1280 \def\MT@undo@patch#1{%
1281   \MT@in@clist{#1}\MT@patches@applied
1282   \ifMT@inlist@
1283     \MT@rem@from@clist{#1}\MT@patches@applied
1284     \@nameuse{MT@patch@undo@@#1}%
1285     \MT@patch@info@undo{#1}%
1286   \else
1287     \MT@warning{Patch `#1' hasn't been applied,\MessageBreak cannot revert it}%
1288   \fi
1289 }
```

Unfortunately, etoolbox is a bit bitchy with hashes in arguments (but who would blame it), so I currently see no other solution than to temporarily reset the catcode of the # character.

```
1290 {\catcode`\#=12
1291 \MT@addto@setup{%
```

Now for the actual patches:

item: \@item, which is a kind of catch-all, as it's internally used for most basic environments (e.g., itemize, enumerate, but also quote, flushleft etc.). For verse (and probably other environments), we also have to patch \everypar ...

- for the base classes

```
1292   \MT@define@patch{item}{%
1293     \MT@append@patch\@item\leftprotrusion
1294     \MT@patch@patch\@item{\everypar{}}{\everypar{\leftprotrusion}}%
```

- beamer patches it too

```
1295     \@ifclassloaded{beamer}
1296       {\MT@append@patch\beamer@@callorigitem\leftprotrusion
1297        \MT@patch@patch\beamer@callorigitem{\ignorespaces}{\ignorespaces\leftprotrusion}}
```

- the simplecv class

```
1298       {\@ifclassloaded{simplecv}
1299         {\MT@append@patch\@topic@item\leftprotrusion}
1300         {}}%
1301   }{}%
```

toc: TOC and friends

```
1302    \MT@define@patch{toc}{%
1303       \MT@append@patch\numberline\leftprotrusion
```

- for the memoir class we also fix the extra leader problem . . .

```
1304       \@ifclassloaded{memoir}
1305         {\MT@append@patch\booknumberline\leftprotrusion
1306          \MT@append@patch\partnumberline\leftprotrusion
1307          \MT@append@patch\chapternumberline\leftprotrusion
1308          \MT@append@patch\cftbookafterpnum\noprotrusion
1309          \MT@append@patch\cftpartafterpnum\noprotrusion
1310          \MT@append@patch\cftchapterafterpnum\noprotrusion
1311          \MT@append@patch\cftsectionafterpnum\noprotrusion
1312          \MT@append@patch\cftsubsectionafterpnum\noprotrusion
1313          \MT@append@patch\cftsubsubsectionafterpnum\noprotrusion
1314          \MT@append@patch\cftparagraphafterpnum\noprotrusion
1315          \MT@append@patch\cftsubparagraphafterpnum\noprotrusion
1316          \MT@append@patch\cftfigureafterpnum\noprotrusion
1317          \MT@append@patch\cfttableafterpnum\noprotrusion}
1318         {}%
1319       }{}%
```

- for the KOMA classes (which load the tocbasic package) we additionally have to switch protrusion back on; this will re-introduce the risk of getting an extra leader dot, but I currently don't see how to easily add \noprotrusion. Therefore, I'll skip this patch for now, saving the joy of wading through scr files for later, all the while waiting for somebody who would understand KOMA better than me.

```
1320    %    \@ifpackageloaded{tocbasic}
1321    %      {\MT@define@patch{toc}
1322    %        {\MT@append@patch\numberline\leftprotrusion
1323    %         \setuptoc{toc}{noprotrusion}%
1324    %         \setuptoc{lof}{noprotrusion}%
1325    %         \setuptoc{lot}{noprotrusion}}
1326    %        {\unsettoc{toc}{noprotrusion}%
1327    %         \unsettoc{lof}{noprotrusion}%
1328    %         \unsettoc{lot}{noprotrusion}}}{}%
```

- (a patch for titletoc would also be worthwhile . . .)

eqnum: equation numbers (with or without amsmath). \eqref relies on \tagform@, so we have to make it use the original definition.

```
1329    \MT@define@patch{eqnum}{%
1330       \@ifclassloaded{IEEEtran}
1331         {\MT@patch@patch\theequationdis{(}{\leftprotrusion{(}}%
1332          \MT@patch@patch\theequationdis{)}{\rightprotrusion{)}}}%
1333          \MT@patch@patch\theIEEEsubequationdis{(}{\leftprotrusion{(}}%
1334          \MT@patch@patch\theIEEEsubequationdis{)}{\rightprotrusion{)}}}}%
1335         {}%
1336       \@ifpackageloaded{amsmath}
1337         {\MT@patch@patch\tagform@{(}{\leftprotrusion{(}}%
1338          \MT@patch@patch\tagform@{)}{\rightprotrusion{)}}%
1339          \MT@patch@patch\eqref{\tagform@}{\@nameuse{MT@patch@saved@\string\tagform@}}}
1340         {\MT@patch@patch\@eqnnum{(}{\leftprotrusion{(}}%
1341          \MT@patch@patch\@eqnnum{)}{\rightprotrusion{)}}}}%
1342       }{}%
```

footnote: footnote text (only visible with block paragraphs)

- hyperref also patches this command (but only if hyperfootnotes=true)

```
1343      \MT@define@patch{footnote}{%
1344        \@ifpackageloaded{hyperref}
1345            {\ifHy@hyperfootnotes\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
1346            \@secondoftwo
1347        {\MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
1348         \MT@patch@patch\@footnotetext{\@empty\ignorespaces}{\@empty\ignorespaces\leftprotrusion}}}
```

- so do the `KOMA` classes (which load `scrkbase`)

```
1349        {\@ifpackageloaded{scrkbase}
1350            {\MT@patch@patch\scr@saved@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}}
```

- base classes

```
1351            {\MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
```

- `memoir` additionally allows footnotes in the margins

```
1352        \@ifclassloaded{memoir}
1353            {\MT@patch@patch\@footnotetext{\foottextfont #1}{\foottextfont\leftprotrusion #1}}
1354            {}}}%
1355      }{}%
```

Finally, execute any redefinitions.

```
1356      \MT@redefined@patches
1357      }}
```

```
1358 ⟨/package⟩
1359 ⟨/package|letterspace⟩
```

1.2 Font setup

We need a font (the `minimal` class doesn't load one).

```
1360 ⟨package⟩\expandafter\ifx\the\font\nullfont\normalfont\fi
```

\MT@setupfont   Setting up a font entails checking for each feature whether it should be applied to the current font (\MT@font).

```
1361 ⟨*pdf-|lua-|xe-⟩
1362 \def\MT@setupfont{%
```

With X∃TEX and LuaTEX the font may not be actually loaded, hence we might see a wrong font (in \MT@get@slot). Therefore, we first load the current font.

```
1363 ⟨xe-|lua-⟩   \MT@font
```

We might have to disable stuff when used together with adventurous packages.

```
1364    \MT@setupfont@hook}
```

This will use a copy of the font (allowing for expansion parameter variation and the use of more than one set of protrusion factors for a font within one paragraph).

```
1365 ⟨pdf-⟩\MT@requires@pdftex7{
1366 ⟨pdf-|lua-⟩\g@addto@macro\MT@setupfont\MT@copy@font
1367 ⟨pdf-⟩}\relax
```

The font properties must be extracted from \MT@font, since the current value of \f@encoding and friends may be wrong!

```
1368 \g@addto@macro\MT@setupfont{%
1369    \MT@exp@two@c\MT@split@name\string\MT@font/\@nil
```

Try to find a configuration file for the current font family.

```
1370    \MT@exp@one@n\MT@find@file\MT@family
1371    \ifx\MT@familyalias\@empty \else
1372      \MT@exp@one@n\MT@find@file\MT@familyalias\fi
```

We have to make sure that \cf@encoding expands to the correct value (for later, in
\MT@get@slot), which isn't the case when \selectfont chooses a new encoding
(this would be done a second later in \selectfont, anyway – three lines, to be
exact). (I think, I do not need this anymore – however, I'm too afraid to remove it.
... Oops, I did it. Let's see whether anybody complains.)

```
1373 % \ifx\f@encoding\cf@encoding\else\@@enc@update\fi
1374 }
```

Tracking has to come first, since it means actually loading a different font.

```
1375 ⟨pdf-⟩\MT@requires@pdftex6
1376 ⟨lua-⟩\MT@requires@luatex3
1377 ⟨pdf-|lua-⟩  {\g@addto@macro\MT@setupfont\MT@tracking}\relax
1378 \g@addto@macro\MT@setupfont{%
1379    \MT@check@font
1380    \ifMT@inlist@
1381 ⟨debug⟩\MT@show@pdfannot2%
1382    \else
1383      \MT@vinfo{Setting up font `\MT@@font'\on@line}%
1384      \MT@info@notracking
```

Now we can begin setting up the font for all features that the current pdfTEX
provides. The following commands are \let to \relax if the respective feature is
disabled via package options.

For versions older than 1.20, protrusion has to be set up first, beginning with
1.20, the order doesn't matter.

```
1385    \MT@protrusion
1386 ⟨pdf-|lua-⟩  \MT@expansion
1387 }
```

Interword spacing and kerning (pdfTEX 1.40).

```
1388 ⟨*pdf-⟩
1389 \MT@requires@pdftex6{
1390 \g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}
1391 }\relax
1392 ⟨/pdf-⟩
```

Disable ligatures (pdfTEX 1.30).

```
1393 ⟨pdf-⟩\MT@requires@pdftex5{
1394 ⟨pdf-|lua-⟩\g@addto@macro\MT@setupfont\MT@noligatures
1395 ⟨pdf-⟩}\relax
1396 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
1397 ⟨debug⟩\MT@show@pdfannot1%
```

Finally, register the font so that we don't set it up anew each time.

```
1398    \MT@register@font
1399    \fi
1400 }
1401 ⟨/pdf-|lua-|xe-⟩
```

\MT@copy@font · The new (1.40.4) \pdfcopyfont command allows expanding a font with different
\MT@copy@font@ · parameters, or to use more than one set of protrusion factors for a given font
within one paragraph. It will be used when we find a context for \SetProtrusion
or \SetExpansion in the preamble, or when the package has been loaded with the
copyfonts option.

```
1402 ⟨*pdf-|lua-⟩
1403 \let\MT@copy@font\relax
1404 ⟨pdf-⟩\MT@requires@pdftex7{
1405 \def\MT@copy@font@{%
```

\MT@font@copy   For every new protrusion and expansion context, we create a new copy.

1406   \xdef\MT@font@copy{\csname\MT@@font/\MT@pr@context/\MT@ex@context\endcsname}%
1407   \expandafter\ifx\MT@font@copy\relax

\MT@font@orig   pdfTeX doesn't allow copying a font that has already been copied and expanded/
letterspaced. Hence, we have to get the original.

1408   \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
1409   \expandafter\ifx\MT@font@orig\relax
1410   \MT@exp@two@c\MT@glet\MT@font@orig\font@name
1411   \else
1412   \MT@exp@two@c\let\font@name\MT@font@orig
1413   \fi
1414 ⟨pdf-⟩   \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name

Even though LuaTeX also provides the primitive from pdfTeX (even renamed to
\copyfont, that is, 'promoted' as per the LuaTeX manual), it is seriously crippled in
that OpenType features will be lost. Therefore, we do not copy the font but load it
anew.

1415 ⟨lua-⟩   \MT@exp@two@c\MT@lua@copyfont\meaning\font@name\@nil
1416 ⟨debug⟩\MT@dinfo1{creating new copy: \MT@font@copy}%

Since it's a new font, we have to remove it from the context lists.

1417   \MT@map@clist@c\MT@active@features{%
1418   \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
1419   \def\@tempa{##1}%
1420   \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@rem@from@list
1421   \fi
1422   }%
1423   \fi
1424   \MT@exp@two@c\let\MT@font\MT@font@copy

We only need the font identifier for letterspacing.

1425   \let\font@name\MT@font@copy

But we have to properly substitute the font after we're done.

1426   \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
1427 }

\MT@rem@from@list

1428 \def\MT@rem@from@list#1{%
1429   \MT@exp@cs\ifx{MT@\@tempa @#1font@list}\relax\else
1430   \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
1431   \MT@font \csname MT@\@tempa @#1font@list\endcsname
1432   \fi
1433 }
1434 ⟨pdf-⟩}}\relax

\MT@lua@copy@font   ⟨#1⟩ and ⟨#2⟩ are 'select' and 'font', respectively, ⟨#3⟩ is the font spec.

1435 ⟨lua-⟩\def\MT@lua@copyfont #1 #2 #3\@nil{%
1436 ⟨lua-⟩ \global\expandafter\font\MT@font@copy=#3\relax}
1437 ⟨/pdf-|lua-⟩

Here's the promised dirty trick   for users of older pdfTeX versions, which works
around the problem that the use of the same font with different expansion para-
meters is prohibited. If you do not want to create a clone of the font setup (this
would require duplicating the `tfm`/`vf` files under a new name, and writing new `fd`
files and `map` entries), you can load a minimally larger font for the paragraph in
question. E.g., for a document typeset in 10 pt:

```
\SetExpansion
  [ stretch = 30,
```

```
        shrink = 60,
        step   =  5 ]
     { encoding = *,
       size = 10.001 }
     { }
  \newcommand{\expandpar}[1]{{%
     \fontsize{10.001}{\baselineskip}\selectfont #1\par}}
  % ...
  \expandpar{This paragraph contains an `unnecessary' widow.}
```

Note that the \expandpar command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the fix-cm package to be able to select fonts in arbitrary sizes. Finally, the reason I suggest to use a larger font, and not a smaller one, is to prevent a different design size being selected.

\MT@fix@fontdimen@six   If \fontdimen 6 is zero, character protrusion, spacing, kerning and tracking won't
\MT@dimen@six   work, and we could skip the settings (for example, the dsfont fonts don't specify this dimension; this is probably a bug – the fourier and newpx/newtx packages have been fixed in the meantime). However, we can fix it ourselves – only tracking still doesn't work (it seems that \letterspacefont uses the \fontdimen 6 from the original font). X͟ETEX doesn't provide an equivalent to \pdffontsize, so we use the nominal size instead.

```
1438 ⟨*pdf-|lua-|xe-⟩
1439 \def\MT@fix@fontdimen@six{%
1440   \ifnum\fontdimen6\MT@font=\z@
1441     \fontdimen6\MT@font=%
1442 ⟨pdf-⟩       \pdffontsize\MT@font
1443 ⟨lua-⟩       \MT@requires@luatex4{\pdffeedback fontsize}{\pdffontsize}\MT@font
1444 ⟨xe-⟩        \MT@size pt
1445     \MT@info{Fixing zero \string\fontdimen 6 for font `\MT@@font'\MessageBreak
1446             (new value: \the\fontdimen6\MT@font)}%
1447 ⟨pdf-⟩      \MT@requires@pdftex8\relax{\MT@glet@nc{\MT@@font-fake6}\@empty}%
1448   \fi
1449   \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1450 }
1451 ⟨/pdf-|lua-|xe-⟩
```

\MT@split@name   Split up the font name (⟨#6⟩ may be a protrusion/expansion context and/or a
\MT@encoding   letterspacing amount). With fontspec we also need to remove its internal instance
\MT@family   counter.
\MT@series
\MT@shape
\MT@size

```
1452 ⟨*package⟩
1453 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
1454   \def\MT@encoding{#1}%
1455   \ifMT@fontspec
1456     \edef\MT@family{\MT@scrubfeature#2()\relax}%
1457   \else
1458     \def\MT@family{#2}%
1459   \fi
1460   \def\MT@series  {#3}%
1461   \def\MT@shape   {#4}%
1462   \def\MT@size    {#5}%
1463   \MT@fix@fontdimen@six
```

\MT@familyalias   Alias family?

```
1464   \MT@ifdefined@n@TF{MT@\MT@family @alias}%
1465     {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
1466     {\let\MT@familyalias\@empty}%
1467 }
```

\MT@scrubfeature   Remove one resp. all feature counters (fontspec).
\MT@scrubfeatures

```
1468 \def\MT@scrubfeature#1(#2)#3\relax{#1}
1469 \def\MT@scrubfeatures#1(#2)#3\relax{%
1470   #1%
1471   \ifx\relax#3\relax\else
1472     \MT@scrubfeatures#3\relax
1473   \fi
1474 }
```

\ifMT@do We check all features of the current font against the lists of the currently active
\MT@feat font set, and set \ifMT@do accordingly.

\MT@maybe@do
```
1475 \newif\ifMT@do
1476 \def\MT@maybe@do#1{%
```

(but only if the feature isn't globally set to false)

```
1477   \csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
```

Begin with setting micro-typography to true for this font. The \MT@checklist@...
tests will set it to false if the property is not in the list. The first non-empty list that
does not contain a match will stop us (except for font).

```
1478     \MT@dotrue
1479     \edef\@tempa{\csname MT@#1@setname\endcsname}%
1480     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1481       \MT@ifdefined@n@TF{MT@checklist@##1}%
1482         {\csname MT@checklist@##1\endcsname}%
1483         {\MT@checklist@{##1}}%
1484       {#1}%
1485     }%
1486   \else
1487     \MT@dofalse
1488   \fi
1489   \ifMT@do
```

\MT@feat stores the current feature.

```
1490     \def\MT@feat{#1}%
1491     \csname MT@set@#1@codes\endcsname
1492   \else
1493     \MT@ifstreq{#1}{tr}%
1494       {\let\MT@info@notracking\MT@info@notracking@}%
1495       {\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}%
1496   \fi
1497 }
```

\MT@info@notracking To defer the message to after the font has actually been logged.

\MT@info@notracking@
```
1498 \let\MT@info@notracking\relax
1499 \def\MT@info@notracking@{\MT@vinfo{... No tracking}}
```

\MT@dinfo@list
```
1500 ⟨debug⟩\def\MT@dinfo@list#1#2#3{\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: #2
1501 ⟨debug⟩  \ifx\\#3\\list empty\else `\@nameuse{MT@#2}' #3 list\fi}}
```

\MT@checklist@ The generic test (⟨#1⟩ is the axis, ⟨#2⟩ the feature, \@tempa contains the set name).

```
1502 \def\MT@checklist@#1#2{%
1503 ⟨!debug⟩  \MT@ifdefined@n@T
1504 ⟨debug⟩   \MT@ifdefined@n@TF
1505       {MT@#2list@#1@\@tempa}{%
```

Begin a (neatly masqueraded) \expandafter orgy to test whether the font attribute
is in the list.

```
1506     \expandafter\MT@exp@one@n\expandafter\MT@in@clist
1507       \csname MT@#1\expandafter\endcsname
1508       \csname MT@#2list@#1@\@tempa\endcsname
1509     \ifMT@inlist@
1510 ⟨debug⟩\MT@dinfo@list{#2}{#1}{in}%
1511       \MT@dotrue
```

```
1512        \else
1513 ⟨debug⟩\MT@dinfo@list{#2}{#1}{not in}%
1514          \MT@dofalse
1515          \expandafter\MT@clist@break
1516        \fi
1517   }%
```

If no limitations have been specified, i.e., the list for a font attribute has not been defined at all, the font should be set up.

```
1518 ⟨debug⟩   {\MT@dinfo@list{#2}{#1}{}}%
1519 }
```

\MT@checklist@family   Also test for the alias font, if the original font is not in the list.

```
1520 \def\MT@checklist@family#1{%
1521 ⟨!debug⟩   \MT@ifdefined@n@T
1522 ⟨debug⟩    \MT@ifdefined@n@TF
1523        {MT@#1list@family@\@tempa}{%
1524      \MT@exp@two@n\MT@in@clist
1525          \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
1526      \ifMT@inlist@
1527 ⟨debug⟩\MT@dinfo@list{#1}{family}{in}%
1528          \MT@dotrue
1529        \else
1530 ⟨debug⟩\MT@dinfo@list{#1}{family}{not in}%
1531          \MT@dofalse
1532        \ifx\MT@familyalias\@empty \else
1533          \MT@exp@two@n\MT@in@clist
1534              \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
1535          \ifMT@inlist@
1536 ⟨debug⟩  \MT@dinfo@list{#1}{family alias}{in}%
1537            \MT@dotrue
1538 ⟨debug⟩\else\MT@dinfo@list{#1}{family alias}{not in}%
1539            \fi
1540          \fi
1541        \fi
1542      \ifMT@do \else
1543          \expandafter\MT@clist@break
1544        \fi
1545   }%
1546 ⟨debug⟩   {\MT@dinfo@list{#1}{family}{}}%
1547 }
```

\MT@checklist@size   Test whether font size is in list of size ranges.

```
1548 \def\MT@checklist@size#1{%
1549 ⟨!debug⟩   \MT@ifdefined@n@T
1550 ⟨debug⟩    \MT@ifdefined@n@TF
1551        {MT@#1list@size@\@tempa}{%
1552      \MT@exp@cs\MT@in@rlist{MT@#1list@size@\@tempa}%
1553      \ifMT@inlist@
1554 ⟨debug⟩\MT@dinfo@list{#1}{size}{in}%
1555          \MT@dotrue
1556        \else
1557 ⟨debug⟩\MT@dinfo@list{#1}{size}{not in}%
1558          \MT@dofalse
1559          \expandafter\MT@clist@break
1560        \fi
1561   }%
1562 ⟨debug⟩   {\MT@dinfo@list{#1}{size}{}}%
1563 }
```

\MT@checklist@font   If the font matches, we skip the rest of the test.

```
1564 \def\MT@checklist@font#1{%
1565 ⟨!debug⟩   \MT@ifdefined@n@T
1566 ⟨debug⟩    \MT@ifdefined@n@TF
1567        {MT@#1list@font@\@tempa}{%
```

Since \MT@font may be appended with context and/or letterspacing specs, we construct the name from the font characteristics.

```
1568        \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1569        \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1570          \@tempb \csname MT@#1list@font@\@tempa\endcsname
1571        \ifMT@inlist@
1572 ⟨debug⟩\MT@dinfo@list{#1}{font}{in}%
1573          \expandafter\MT@clist@break
1574        \else
1575 ⟨debug⟩\MT@dinfo@list{#1}{font}{not in}%
1576          \MT@dofalse
1577        \fi
1578   }%
1579 ⟨debug⟩   {\MT@dinfo@list{#1}{font}{}}%
1580 }
```

1.2.1 Protrusion

\ifMT@nofamily   Info for settings that are not family-specific. (Warnings seem to be too irritating.) The switch is set in \MT@next@listname.

```
1581 \newif\ifMT@nofamily
```

\MT@protrusion   Set up for protrusion?

```
1582 \def\MT@protrusion{\MT@maybe@do{pr}}
1583 ⟨/package⟩
```

\MT@set@pr@codes   This macro is called by \MT@setupfont, and does all the work for setting up a font for protrusion.

```
1584 ⟨*pdf-|lua-|xe-|show⟩
1585 ⟨show⟩\def\MTS@show@pr
1586 ⟨pdf-|lua-|xe-⟩\def\MT@set@pr@codes
1587   {%
1588 ⟨pdf-|lua-|xe-⟩ \MT@nofamilyfalse
```

Check whether and if, which list should be applied to the current font. If family-specific settings don't exist, we write it to the log (for each encoding).

```
1589 ⟨show⟩   \MTS@printtext{Protrusion settings for font `\texttt{\MT@@font}':}\\
1590      \MT@if@list@exists{%
1591 ⟨*pdf-|lua-|xe-⟩
1592        \ifMT@nofamily
1593          \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
1594            \MT@info@nl{Loading generic protrusion settings for font family\MessageBreak
1595                       `\MT@family' (encoding: \MT@encoding).\MessageBreak
1596                       For optimal results, create family-specific settings.\MessageBreak
1597                       See the microtype manual for details}%
1598            \MT@glet@nc{\MT@encoding-\MT@family-settings}\@empty
1599          }%
1600        \fi
1601 ⟨/pdf-|lua-|xe-⟩
1602 ⟨show⟩      \MTS@printtext{First matching list is for `\texttt{\@tempa}':\\\texttt{\MT@pr@c@name}}%
1603      \MT@get@opt
1604      \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

```
1605      \MT@get@inh@list
```

Set an input encoding?

```
1606      \MT@set@inputenc{c}%
```

Load additional lists?

```
1607      \MT@load@list\MT@pr@c@name
1608      \MT@set@listname
```

Load the main list.

```
1609      \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
1610      \expandafter\MT@set@codes\@tempc,\relax,%
1611 ⟨show⟩    \vrule width 4cm height .5pt \\
1612 ⟨show⟩    \MTS@printtext{End of list `\texttt{\MT@pr@c@name}'}\\[.5em]
1613 ⟨show⟩    \MT@ifdefined@c@T\MT@pr@inh@name{%
1614 ⟨show⟩      \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @prefixes}{%
1615 ⟨show⟩        \par \MTS@printtext{(with prefixes:)}%
1616 ⟨show⟩        \@tempcntb=\z@
```

Set unconditional heirs.

```
1617      \MT@set@pr@prefixheirs
1618 ⟨show⟩      }}%
1619 ⟨show⟩    \ifShowMissingGlyphs\MTS@show@missing\fi
1620    }%
1621 ⟨show⟩  {\MTS@printtext{NOT DEFINED}%
1622    \MT@reset@pr@codes
1623 ⟨show⟩  }\par
1624 }
```

\MT@set@all@pr Set all protrusion codes of the font.

```
1625 ⟨*pdf-|lua-|xe-⟩
1626 \def\MT@set@all@pr#1#2{%
1627 ⟨debug⟩\MT@dinfo@nl{3}{-- lp/rp: setting all to #1/#2}%
1628    \let\MT@temp\@empty
1629    \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1}}%
1630    \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2}}%
1631    \MT@do@font\MT@temp
1632 }
```

\MT@reset@pr@codes@ All protrusion codes are zero for new fonts. However, if we have to reload the font
\MT@reset@pr@codes due to different contexts, we have to reset them. This command will be changed by
\microtypecontext if necessary.

```
1633 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
1634 \let\MT@reset@pr@codes\relax
```

\MT@the@pr@code If the font is letterspaced, we have to add half the letterspacing amount to the
\MT@the@pr@code@tr margin kerns. This will be activated in \MT@set@tr@codes.

```
1635 \def\MT@the@pr@code{\@tempcntb}
1636 ⟨*pdf-|lua-⟩
1637 ⟨pdf-⟩\MT@requires@pdftex6
1638 ⟨lua-⟩\MT@requires@luatex3
1639    {\def\MT@the@pr@code@tr{%
1640      \numexpr\@tempcntb+\MT@letterspace@/2\relax
1641    }
1642 }\relax
1643 ⟨/pdf-|lua-⟩
```

\MT@set@codes Split up the values and set the codes.

```
1644 \def\MT@set@codes#1,{%
1645    \ifx\relax#1\@empty\else
1646      \MT@split@codes #1==\relax
1647      \expandafter\MT@set@codes
1648    \fi
1649 }
```

\MT@split@codes The keyval package would remove spaces here, which we needn't do since
\SetProtrusion ignores spaces in the protrusion list anyway. \MT@get@char@unit
may mean different things.

```
1650 \def\MT@split@codes#1=#2=#3\relax{%
1651    \def\@tempa{#1}%
1652    \ifx\@tempa\@empty \else
1653      \MT@get@slot
```

```
1654 ⟨pdf-|lua-⟩    \ifnum\MT@char > \m@ne
1655 ⟨xe-⟩    \ifx\MT@char\@empty \else
1656       \MT@get@char@unit
1657       \csname MT@\MT@feat @split@val\endcsname#2\relax
1658     \fi
1659   \fi
1660 }
```

\MT@pr@split@val

```
1661 \def\MT@pr@split@val#1,#2\relax
1662 ⟨/pdf-|lua-|xe-⟩
1663 ⟨show⟩\def\MTS@pr@split@val#1,#2\relax
1664   {\def\@tempb{#1}%
1665   \MT@ifempty\@tempb
1666 ⟨pdf-|lua-|xe-⟩    \relax
1667 ⟨show⟩   {\MTS@lp@=\z@ \let\MTS@lpcode\@empty}%
1668   {\MT@scale@to@em
1669 ⟨pdf-|lua-|xe-⟩    \lpcode\MT@font\MT@char=\MT@the@pr@code
1670 ⟨show⟩     \MTS@lp@=\dimexpr\@tempcntb em/1000\relax\relax
1671 ⟨show⟩     \edef\MTS@lpcode{[\@tempb] \the\@tempcntb/\the\MTS@lp@}%
1672 ⟨debug⟩\MT@dinfo@nl{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
1673   }%
1674   \def\@tempb{#2}%
1675   \MT@ifempty\@tempb
1676 ⟨pdf-|lua-|xe-⟩    \relax
1677 ⟨show⟩   {\MTS@rp@=\z@ \let\MTS@rpcode\@empty}%
1678   {\MT@scale@to@em
1679 ⟨pdf-|lua-|xe-⟩    \rpcode\MT@font\MT@char=\MT@the@pr@code
1680 ⟨show⟩     \MTS@rp@=\dimexpr\@tempcntb em/1000\relax\relax
1681 ⟨show⟩     \edef\MTS@rpcode{[\@tempb] \the\@tempcntb/\the\MTS@rp@}%
1682 ⟨debug⟩\MT@dinfo@nl{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char: [#2]}%
1683   }%
1684 ⟨show⟩  \llap{\MTS@show@char@pr\MT@char\quad}%
1685 ⟨show⟩  \parbox[b][][b]{3.5cm}{\MTS@printtext{%
1686 ⟨show⟩      \footnotesize\makebox[.4cm][l]{L:} \MT@ifempty{\MTS@lpcode}{---}{\MTS@lpcode}\\
1687 ⟨show⟩            \makebox[.4cm][l]{R:} \MT@ifempty{\MTS@rpcode}{---}{\MTS@rpcode}}}%
1688 ⟨show⟩  \parbox[t][][t]{\dimexpr\textwidth-3.5cm}{%
```

Now we can set the values for the inheriting characters. Their slot numbers are saved in the macro \MT@inh@⟨*list name*⟩@⟨*slot number*⟩@.

```
1689   \MT@ifdefined@c@T\MT@pr@inh@name{%
1690     \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @\MT@char @}{%
1691       \MT@exp@cs\MT@map@tlist@c
1692         {MT@inh@\MT@pr@inh@name @\MT@char @}%
1693 ⟨pdf-|lua-|xe-⟩          \MT@set@pr@heirs
1694 ⟨show⟩       \MTS@show@char@pr
1695     }%
1696   }%
1697 ⟨show⟩  }\newline
1698 }
1699 ⟨*pdf-|lua-|xe-⟩
```

\MT@scale@to@em   Since pdfTEX version 0.14h, we have to adjust the protrusion factors (i.e., convert numbers from thousandths of character width to thousandths of an em of the font). We have to do this *before* setting the inheriting characters, so that the latter inherit the absolute value, not the relative one if they have a differing width (e.g., the 'ff' ligature). Unlike protcode.tex and pdfcprot, we do not calculate with \lpcode resp. \rpcode, since this would disallow protrusion factors larger than the character width (since \[lr]pcode's limit is 1000). Now, the maximum protrusion is 1 em of the font.

The unit is in \MT@count, the desired factor in \@tempb, and the result will be returned in \@tempcntb.

```
1700 ⟨pdf-⟩\MT@requires@pdftex3{
1701 \def\MT@scale@to@em{%
1702   \@tempcntb=\MT@count\relax
```

For really huge fonts (100 pt or so), an arithmetic overflow could occur with vanilla
TeX. Using e-TeX, this can't happen, since the intermediate value is 64 bit, which
could only be reached with a character width larger than \maxdimen.

```
1703   \MT@scale\@tempcntb \@tempb \MT@dimen@six
1704   \ifnum\@tempcntb=\z@ \else
1705     \MT@scale@factor
1706   \fi
1707 }
```

\MT@get@charwd   Get the width of the character. When using e-TeX, we can employ \fontcharwd
instead of building scratch boxes.

```
1708 \def\MT@get@charwd{%
1709 ⟨*pdf-⟩
1710 ^^X  \MT@count=\fontcharwd\MT@font\MT@char\relax
1711 ^^Q  \setbox\z@=\hbox{\MT@font \char\MT@char}%
1712 ^^Q  \MT@count=\wd\z@
1713 ⟨/pdf-⟩
1714 ⟨lua-⟩  \MT@count=\fontcharwd\MT@font\MT@char\relax
```

\MT@char contains a slot number (legacy fonts), a Unicode number, or a glyph name
(if \MT@char@ is negative).

```
1715 ⟨*xe-⟩
1716   \ifnum\MT@char@<\z@
1717     \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}%
1718     \MT@count=\wd\z@
1719   \else
1720     \MT@count=\fontcharwd\MT@font\MT@char@\relax
1721   \fi
1722 ⟨/xe-⟩
1723   \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1724 }
```

For letterspaced fonts, we have to subtract the letterspacing amount from the
characters' widths. The protrusion amounts will be adjusted in \MT@set@pr@codes.
The letterspaced font is already loaded so that 1 em = \fontdimen 6.

```
1725 ⟨*pdf-⟩
1726 \MT@requires@pdftex6{
1727   \g@addto@macro\MT@get@charwd{%
1728     \MT@ifdefined@c@T\MT@letterspace@
1729       {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1730   }
1731 }\relax
1732 }{
```

No adjustment with versions 0.14f and 0.14g.

```
1733 \def\MT@scale@to@em{%
1734   \MT@count=\@tempb\relax
1735   \ifnum\MT@count=\z@ \else
1736     \MT@scale@factor
1737   \fi
1738 }
```

We need this in \MT@warn@code@too@large (neutralised).

```
1739 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1740 }
1741 ⟨/pdf-⟩
1742 ⟨/pdf-|lua-|xe-⟩
1743 ⟨/pdf-|lua-|xe-|show⟩
```

`\MT@get@font@dimen`   For the space unit.

```
1744 ⟨*package⟩
1745 \def\MT@get@font@dimen#1{%
1746   \ifnum\fontdimen#1\MT@font=\z@
1747     \MT@warning@nl{Font `\MT@@font' does not specify its\MessageBreak
1748       \@backslashchar fontdimen #1 (it's zero)!\MessageBreak
1749       You should use a different `unit' for \MT@curr@list@name}%
1750   \else
1751     \MT@count=\fontdimen#1\MT@font
1752   \fi
1753 }
```

`\MT@info@missing@char`   Info about missing characters, or characters with zero width.

```
1754 \def\MT@info@missing@char{%
1755   \MT@info@nl{Character `\the\MT@toks'
1756 ^^X    \ifnum\MT@char@<\z@ is missing\else
1757 ^^X      \iffontchar\MT@font\MT@char@
1758            has a width of 0pt
1759 ^^X      \else is missing\fi\fi
1760 ^^Q    \MessageBreak (it's probably missing)
1761     \MessageBreak in font `\MT@@font'.\MessageBreak
1762     Ignoring protrusion settings for this character}%
1763 }
```

`\MT@scale@factor`   Furthermore, we might have to multiply with a factor.

```
1764 \def\MT@scale@factor{%
1765   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1766     \expandafter\MT@scale\expandafter \@tempcntb
1767       \csname MT@\MT@feat @factor@\endcsname \@m
1768   \fi
1769   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1770     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1771   \else
1772     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1773       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1774     \fi
1775   \fi
1776 }
```

`\MT@warn@code@too@large`   Type out a warning if a chosen protrusion factor is too large after the conversion. As a special service, we also type out the maximum amount that may be specified in the configuration.

```
1777 \def\MT@warn@code@too@large#1{%
1778   \@tempcnta=#1\relax
1779   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1780     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1781       \@m \csname MT@\MT@feat @factor@\endcsname
1782   \fi
1783   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1784   \MT@warning@nl{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1785     is too large for character\MessageBreak
1786     `\the\MT@toks' in \MT@curr@list@name.\MessageBreak
1787     Setting it to the maximum of \number\@tempcnta}%
1788   \@tempcntb=#1\relax
1789 }
```

`\MT@get@opt`   The optional argument to the configuration commands (except for \SetExpansion and \SetTracking, which are being dealt with in \MT@get@ex@opt and \MT@get@tr@opt, resp.).

```
1790 \def\MT@get@opt{%
1791   \MT@set@listname
```

`\MT@pr@factor@`
`\MT@sp@factor@`
`\MT@kn@factor@`   Apply a factor?

```
1792   \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
1793     \MT@let@nn{MT@\MT@feat @factor@}
1794        {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1795     \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1796                    \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1797   }{%
1798     \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1799   }%
```

\MT@pr@unit@   The unit can only be evaluated here, since it might be font-specific. If it's \@empty,
\MT@sp@unit@   it's relative to character widths, if it's −1, relative to space dimensions.
\MT@kn@unit@
```
1800   \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1801     \MT@let@nn{MT@\MT@feat @unit@}%
1802        {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1803     \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1804       \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1805                     relative to character widths}%
1806     \else
1807       \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1808         \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1809                       relative to width of space}%
1810       \fi
1811     \fi
1812   }{%
1813     \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1814   }%
```

\MT@get@space@unit   The codes are either relative to character widths, or to a fixed width. For spacing
\MT@get@char@unit   and kerning lists, they may also be relative to the width of the interword glue. Only
                       the setting from the top list will be taken into account.
```
1815   \let\MT@get@char@unit\relax
1816   \let\MT@get@space@unit\@gobble
1817   \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1818     \let\MT@get@char@unit\MT@get@charwd
1819   \else
1820     \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1821       \let\MT@get@space@unit\MT@get@font@dimen
1822     \else
1823       \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1824     \fi
1825   \fi
```

Preset all characters? If so, we surely don't need to reset, too.
```
1826   \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
1827     \csname MT@preset@\MT@feat\endcsname
1828     \MT@let@nc{MT@reset@\MT@feat @codes}\relax
1829   }%
1830 }
```

\MT@get@unit   If unit contains an em or ex, we use the corresponding \fontdimen to obtain the
\MT@get@unit@   real size. Simply converting the em into points might give a wrong result, since
                 the font probably isn't set up yet, so that these dimensions haven't been updated,
                 either.
```
1831 \def\MT@get@unit#1{%
1832   \expandafter\MT@get@unit@#1 e!\@nil
1833   \ifx\x\@empty\else\let#1\x\fi
1834   \@defaultunits\@tempdima#1 pt\relax\@nnil
1835   \ifdim\@tempdima=\z@
1836     \MT@warning@nl{%
1837       Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
1838       width. Setting factors of list `\@nameuse{MT@\MT@feat @c@name}'\MessageBreak
1839       relative to character widths instead}%
1840     \let#1\@empty
```

```
1841        \let\MT@get@char@unit\MT@get@charwd
1842    \else
1843      \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
1844                       to \the\@tempdima}%
1845      \MT@count=\@tempdima\relax
1846    \fi
1847 }
1848 \def\MT@get@unit@#1e#2#3\@nil{%
1849    \ifx\\#3\\\let\x\@empty \else
1850      \if m#2%
1851        \edef\x{#1\fontdimen6\MT@font}%
1852      \else
1853        \if x#2%
1854          \edef\x{#1\fontdimen5\MT@font}%
1855        \fi
1856      \fi
1857    \fi
1858 }
```

\MT@set@inputenc   The configurations may be under the regime of an input encoding.

```
1859 \def\MT@set@inputenc#1{%
```

\MT@cat   We remember the current category (c or inh), in case of warnings later.

```
1860    \def\MT@cat{#1}%
```

```
1861    \edef\@tempa{MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}%
1862    \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1863 }
```

\MT@set@inputenc@   More recent versions of inputenc remember the current encoding, so that we can test whether we really have to load the encoding file.

```
1864 \MT@addto@setup{%
1865    \@ifpackageloaded{inputenc}{%
1866      \@ifpackagelater{inputenc}{2006/02/22}{%
1867        \def\MT@set@inputenc@{%
1868          \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1869            \MT@load@inputenc
1870        }%
1871      }{%
1872        \let\MT@set@inputenc@\MT@load@inputenc
1873      }%
1874    }{%
1875      \def\MT@set@inputenc@{%
1876        \MT@warning@nl{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1877            \MessageBreak package isn't loaded. Ignoring input encoding}%
1878      }%
1879    }%
1880 }
```

\MT@load@inputenc   Set up normal catcodes, since, e.g., listings would otherwise want to actually typeset the inputenc file when it is being loaded inside a listing.

```
1881 \def\MT@load@inputenc{%
1882    \MT@cfg@catcodes
1883 ⟨debug⟩\MT@dinfo@nl{1}{loading input encoding: \@nameuse{\@tempa}}%
1884    \inputencoding{\@nameuse{\@tempa}}%
1885 }
```

\MT@set@pr@heirs   Set the inheriting characters.

```
1886 \def\MT@set@pr@heirs#1{%
1887    \lpcode\MT@font #1=\lpcode\MT@font\MT@char\relax
1888    \rpcode\MT@font #1=\rpcode\MT@font\MT@char\relax
1889 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
1890 ⟨debug⟩\MT@dinfo@nl{4}{;;; lp/rp (#1): \number\lpcode\MT@font\MT@char/%
1891 ⟨debug⟩                                    \number\rpcode\MT@font\MT@char}%
1892 }
```

\MT@set@pr@prefixheirs   Inheriting characters that have been specified in a prefixed list.

```
1893 \def\MT@set@pr@prefixheirs{%
1894   \MT@ifdefined@c@T\MT@pr@inh@name{%
1895     \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @prefixes}{%
1896       \MT@exp@cs\MT@map@tlist@c
1897         {MT@inh@\MT@pr@inh@name @prefixes}%
1898         \MT@set@pr@prefixes
1899     }%
1900   }%
1901 }
1902 ⟨/package⟩
```

\MT@set@pr@prefixes   Add charwidth(inheriting char)-charwidth(base char) to either left or right
\MT@set@pr@prefixes@   side or half the amount to both sides. For X∃TEX, we may have to translate to glyph
                        numbers because \fontcharwd doesn't have the nice feature of understanding the
                        'U' or '/' prefixes.

```
1903 ⟨*pdf-|lua-|xe-|show⟩
1904 ⟨pdf-|lua-|xe-⟩\def\MT@set@pr@prefixes#1{\MT@set@pr@prefixes@#1}
1905 ⟨pdf-|lua-|xe-⟩\def\MT@set@pr@prefixes@#1#2#3#4%
1906 ⟨show⟩\def\MTS@set@pr@prefixes@#1#2#3#4%
1907   {%
1908 ⟨show⟩   \MTS@lp@=\z@ \MTS@rp@=\z@
1909 ⟨show⟩   \ifnum#1=\@tempcntb \else
1910 ⟨show⟩     \par\leavevmode
1911 ⟨show⟩     \llap{\MTS@show@char@pr{#1} \MTS@printtext{=} }%
1912 ⟨show⟩   \fi
1913 ⟨*xe-⟩
1914   \edef\@tempa{\expandafter\ifx\@car#1\@nil U\@gobble#1\else\number\XeTeXglyphindex"#1" \fi}%
1915   \edef\@tempb{\expandafter\ifx\@car#2\@nil U\@gobble#2\else\number\XeTeXglyphindex"#2" \fi}%
1916 ⟨/xe-⟩
1917   \@tempcnta=\z@
1918   \ifnum#3>\z@
1919     \@tempcnta=\numexpr
1920 ⟨pdf-|lua-|show⟩       (\fontcharwd\MT@font#2-\fontcharwd\MT@font#1)%
1921 ⟨xe-⟩       (\fontcharwd\MT@font\@tempb-\fontcharwd\MT@font\@tempa)%
1922       *#3/\MT@dimen@six\relax
1923   \fi
1924 ⟨pdf-|lua-|xe-⟩  \lpcode\MT@font #2=\numexpr\lpcode\MT@font#1+\@tempcnta\relax
1925 ⟨show⟩   \MTS@lp@=\dimexpr\numexpr\lpcode\MT@font#1+\@tempcnta\relax em/1000\relax
1926   \@tempcnta=\z@
1927   \ifnum#4>\z@
1928     \@tempcnta=\numexpr
1929 ⟨pdf-|lua-|show⟩       (\fontcharwd\MT@font#2-\fontcharwd\MT@font#1)%
1930 ⟨xe-⟩       (\fontcharwd\MT@font\@tempb-\fontcharwd\MT@font\@tempa)%
1931       *#4/\MT@dimen@six\relax
1932   \fi
1933 ⟨pdf-|lua-|xe-⟩  \rpcode\MT@font #2=\numexpr\rpcode\MT@font#1+\@tempcnta\relax
1934 ⟨show⟩   \MTS@rp@=\dimexpr\numexpr\rpcode\MT@font#1+\@tempcnta\relax em/1000\relax
1935 ⟨debug⟩\MT@dinfo@nl{2}{-- (prefix) heir of #1: #2}%
1936 ⟨debug⟩\MT@dinfo@nl{4}{;;; lp/rp (#2): \number\lpcode\MT@font#2/%
1937 ⟨debug⟩                                 \number\rpcode\MT@font#2}%
1938 ⟨show⟩   \MTS@show@char@pr{#2}%
1939 ⟨show⟩   \@tempcntb=#1\relax
1940 }
1941 ⟨/pdf-|lua-|xe-|show⟩
```

\MT@preset@pr   Preset characters. Presetting them relative to their widths is not allowed.
\MT@preset@pr@

```
1942 ⟨*package⟩
1943 \def\MT@preset@pr{%
1944   \expandafter\expandafter\expandafter\MT@preset@pr@
1945     \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1946 }
1947 \def\MT@preset@pr@#1,#2\@nil{%
1948   \ifx\MT@pr@unit@\@empty
```

```
1949      \MT@warn@preset@towidth{pr}%
1950      \let\MT@preset@aux\MT@preset@aux@factor
1951    \else
1952      \def\MT@preset@aux{\MT@preset@aux@space2}%
1953    \fi
1954    \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1955    \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1956    \MT@set@all@pr\@tempa\@tempb
1957 }
```

\MT@preset@aux   Auxiliary macro for presetting. Store value ⟨#1⟩ in macro ⟨#2⟩.

\MT@preset@aux@factor
\MT@preset@aux@space

```
1958 \def\MT@preset@aux@factor#1#2{%
1959    \@tempcntb=#1\relax
1960    \MT@scale@factor
1961    \edef#2{\number\@tempcntb}%
1962 }
1963 \def\MT@preset@aux@space#1#2#3{%
1964    \def\@tempb{#2}%
1965    \MT@get@space@unit#1%
1966    \MT@scale@to@em
1967    \edef#3{\number\@tempcntb}%
1968 }
```

\MT@warn@preset@towidth

```
1969 \def\MT@warn@preset@towidth#1{%
1970    \MT@warning@nl{%
1971      Cannot preset characters relative to their widths\MessageBreak
1972      for \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'.
1973      Presetting them\MessageBreak relative to 1em instead}%
1974 }
```

\noprotrusion   This command may be used to inhibit protrusion on either side. It's part of LaTeX
				since 2018-12-01. We provide it for older releases.

```
1975 \MT@ifdefined@c@TF\noprotrusion\relax{
1976    \DeclareRobustCommand\noprotrusion{\leavevmode\kern-\p@\kern\p@}
1977 }
```

\MT@tempbox

```
1978 \newbox\MT@tempbox
```

\leftprotrusion   This command may be used to add protrusion on the left hand side. We try to
				reconstruct the next glyph (possibly a ligature). [3]

```
1979 \DeclareRobustCommand\leftprotrusion{%
1980    \MT@toks{}%
1981    \MT@prot@get@firstchar
1982 }
```

\MT@prot@l   This probably doesn't need to be \long any longer.

```
1983 \long\def\MT@prot@l#1{%
1984    \MT@get@prot{#1}{left}%
1985    #1%
1986 }
```

\rightprotrusion   Unfortunately, there's no way to retrieve anything that's already been typeset, so
				the counterpart cannot be defined symmetrically.

```
1987 \DeclareRobustCommand\rightprotrusion{\MT@prot@r}
```

\MT@prot@r

```
1988 \long\def\MT@prot@r#1{%
1989    {#1}%
1990    \MT@get@prot{#1}{right}%
```

---

3   LuaTeX offers the command \protrusionboundary, which could potentially be very helpful here, but it
	doesn't seem to do what it promises (not even the example from the manual works as advertised).

1991 }

\MT@get@prot   Typeset the text inside a box and get the left and right margin kerns. We add an
extra \vbox in case we're inside a tabular. \@newlistfalse is meant to make \\
work in centering etc.   We set various penalties to zero to allow linebreaking, and
don't bother if the split box is overfull (but shouldn't we? – after all, that's how the
penalties bug was discovered . . .).

\MT@ckpt   We also reset the counters and disable writing to auxiliary files.

\MT@prot@hook   Furthermore, we have a hook for compatibility fixes.

```
1992 \let\MT@prot@hook\@empty
1993 \long\def\MT@get@prot#1#2{%
1994   \begingroup
1995     \def\@elt##1{\global\value{##1}\the\value{##1}\relax}%
1996     \edef\MT@ckpt{\cl@@ckpt}%
1997     \let\@elt\relax
1998     \@fileswfalse
1999     \setbox\MT@tempbox\vbox{%
2000       \everypar{}%
2001       \parfillskip=\z@skip
2002       \hbadness\@M
2003       \clubpenalty\z@
2004       \widowpenalty\z@
2005       \interlinepenalty\z@
2006       \@newlistfalse
2007       \MT@prot@hook
2008       \noindent #1}%
2009     \vbadness=\@M
2010     \splittopskip=\z@
2011     \vfuzz=\maxdimen
2012     \setbox\MT@tempbox\vbox{%
2013       \ifvbox\MT@tempbox
2014         \global\setbox\MT@tempbox=\vsplit\MT@tempbox to \normalbaselineskip
2015         \unvbox\MT@tempbox
2016         \global\setbox\MT@tempbox=\lastbox
2017       \fi
2018     }%
2019     \MT@ckpt
2020   \endgroup
2021   \leavevmode
2022   \ifhbox\MT@tempbox
2023     \@tempdima=\@nameuse{#2marginkern}\MT@tempbox\relax
2024     \expandafter\ifdim\@tempdima=\z@ \else
2025       \MT@vinfo{|<< adding #2 margin kern for `#1':\MessageBreak
2026         \the\@tempdima \on@line}%
2027       \kern\@tempdima
2028     \fi
2029   \fi
2030 }
```

\MT@prot@ifx   Test next token.

```
2031 \def\MT@prot@ifx#1{%
2032   \ifx\MT@prot@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2033 }
```

\MT@prot@ifcat   Test catcode of next token.

```
2034 \def\MT@prot@ifcat#1{%
2035   \ifcat#1\noexpand\MT@prot@next\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2036 }
```

\MT@prot@ifmacro   Test whether ⟨#1⟩ is a macro or an active character that does not take an argument.
\MT@prot@ifmacro@   (inspired by Joseph Wright). Only works with e-TEX.
\MT@prot@ifmacro@@

```
2037 ^^X\def\MT@prot@ifmacro{%
2038 ^^X  \expandafter\MT@prot@ifmacro@\meaning\MT@prot@next\@nil
```

```
2039 ^^X}
2040 ^^X\edef\MT@prot@ifmacro@#1\@nil{%
2041 ^^X    \noexpand\MT@prot@ifmacro@@#1{}{}\detokenize{macro:->}\noexpand\@nil
2042 ^^X}
2043 ^^X\edef\MT@temp{%
2044 ^^X    \def\noexpand\MT@prot@ifmacro@@##1\detokenize{macro:->}##2\noexpand\@nil{%
2045 ^^X        \noexpand\ifx\relax##1\relax
2046 ^^X        \unexpanded{\expandafter\@firstofone\else\expandafter\@gobble\fi}%
2047 ^^X    }%
2048 ^^X}
2049 ^^X\MT@temp
2050 ^^Q\let\MT@prot@ifmacro\@gobble
```

\MT@prot@iffirstcmd   Test for the first command.

```
2051 \def\MT@prot@iffirstcmd#1{%
2052   \ifx\relax#1\relax\expandafter\@secondoftwo\else
2053     \MT@exp@two@c\ifx\@car\MT@prot@next\relax\@empty\@nil#1%
2054       \expandafter\expandafter\expandafter\@firstoftwo
2055     \else
2056       \expandafter\expandafter\expandafter\@secondoftwo
2057     \fi
2058   \fi
2059 }
```

\MT@prot@iflicrcmd   Fun with LICR: Test if the first command of the third command of the first command
                        is \@text@composite, in which case also grab the next token, otherwise it should
                        be a text command.

```
2060 \def\MT@getthird#1#2#3#4\@nil{#3}
2061 \def\MT@prot@iflicrcmd#1{%
2062   \MT@exp@cs\MT@prot@iffirstcmd{#1-cmd}{%
2063     \expandafter\expandafter\expandafter\let
2064       \expandafter\expandafter\expandafter\@tempa
2065       \expandafter\MT@getthird\MT@prot@next\relax\@nil
2066     \MT@exp@two@c\ifx\@car\@tempa\relax\@nil\@text@composite
2067       \def\MT@temp*##1##2{\MT@prot@l{##1##2}}%
2068     \else
2069       \def\MT@temp*##1{\MT@prot@l{##1}}%
2070     \fi
2071     \@gobble
2072   }\@firstofone
2073 }
```

\MT@prot@group   If we have a group, we inject \MT@prot@get@firstchar at the beginning again and
                    don't bother about the rest. This still allows, e.g., \verb, verbatim or lstlistings
                    material.

```
2074 \def\MT@prot@group#{\bgroup\afterassignment\MT@prot@get@firstchar\let\MT@temp= }
```

\MT@prot@get@firstchar   Scan token by token.
\MT@prot@get@nextchar
```
2075 \def\MT@prot@get@firstchar{\futurelet\MT@prot@next\MT@prot@get@first@char}
2076 \def\MT@prot@get@nextchar{\futurelet\MT@prot@next\MT@prot@get@next@char}
```

\MT@prot@get@first@char   If next char is {, treat what follows as an argument, else continue until we find a
                             beginning char.

```
2077 \def\MT@prot@get@first@char{%
2078   \MT@prot@ifcat\bgroup{%
2079     \def\MT@temp*{\MT@prot@group}%
2080   }{%
2081     \MT@prot@ifx\ignorespaces{%
2082       \def\MT@temp*##1{\MT@prot@get@firstchar}%
2083     }{%
2084       \MT@prot@ifx\relax{%
2085         \def\MT@temp*##1{\MT@prot@get@firstchar}%
2086       }{%
```

```
2087        \MT@prot@ifx\@sptoken{%
2088          \def\MT@temp* {\MT@prot@get@firstchar}%
2089        }{%
```

But only add it if it's a letter or a character, …

```
2090          \MT@prot@ifcat{a}{%
2091            \def\MT@temp*{\MT@prot@addtoken@first}%
2092          }{%
2093            \MT@prot@ifcat{!}{%
2094              \def\MT@temp*{\MT@prot@addtoken@first}%
2095            }{%
```

… or a command/active char whose first command is one of the below.

```
2096                \def\MT@temp*{}%
2097                \MT@prot@ifmacro{%
2098                  \MT@prot@iffirstcmd\UTFviii@two@octets{%
2099                    \def\MT@temp*##1##2{\MT@prot@l{##1##2}}%
2100                }{%
2101                  \MT@prot@iffirstcmd\UTFviii@three@octets{%
2102                    \def\MT@temp*##1##2##3{\MT@prot@l{##1##2##3}}%
2103                }{%
2104                  \MT@prot@iffirstcmd\UTFviii@four@octets{%
2105                    \def\MT@temp*##1##2##3##4{\MT@prot@l{##1##2##3##4}}%
2106                }{%
2107                  \MT@prot@iflicrcmd{T1}{%
2108                    \MT@prot@iflicrcmd{TU}{%
2109                      \MT@prot@iflicrcmd{LY1}{%
2110                        \MT@prot@iflicrcmd{OT1}{%
2111                          \MT@prot@iflicrcmd{T2A}\relax % should we add more encodings?
2112                        }%
2113                      }%
2114                    }%
2115                  }%
2116                }%
2117              }%
2118            }%
2119          }%
2120        }%
2121      }%
2122    }%
2123  }%
2124  }%
2125  }%
2126  \MT@temp*%
2127 }
```

`\MT@prot@ifx`   Continue if letter or other.

```
2128 \def\MT@prot@get@next@char{%
2129   \def\MT@temp*{\MT@prot@addtoken@next}%
2130   \MT@prot@ifcat{a}\relax{%
2131     \MT@prot@ifcat{!}\relax{%
2132       \def\MT@temp*{\MT@prot@l{\the\MT@toks}}%
2133     }%
2134   }%
2135   \MT@temp*%
2136 }
```

`\MT@prot@addtoken@first`   Begin filling toks.

```
2137 \def\MT@prot@addtoken@first#1{%
2138   \edef\MT@temp{\MT@toks={\the\MT@toks\noexpand#1}}\MT@temp
2139   \MT@prot@get@nextchar
2140 }
2141 ⟨/package⟩
```

`\MT@prot@addtoken@next`   Add token to our toks and test whether we've seen enough (ligature completed).

For luatex, we have to jump through another hoop (i.e., box), because, contrary to the manual, \lastnodetype isn't really compatible.

```
2142 ⟨*pdf-|lua-|xe-⟩
2143 \def\MT@prot@addtoken@next#1{%
2144   \edef\MT@temp{\MT@toks={\the\MT@toks\noexpand#1}}\MT@temp
2145   \setbox\MT@tempbox\hbox{\the\MT@toks
2146 ⟨pdf-|xe-⟩     \relax
2147 ⟨lua-⟩   }\setbox\MT@tempbox\hbox{\unhbox\MT@tempbox
2148     \ifnum\lastnodetype=7 \aftergroup\@firstoftwo\else\aftergroup\@secondoftwo\fi}%
2149       \MT@prot@get@nextchar
2150       {\MT@prot@l{\the\MT@toks}}%
2151 }
2152 ⟨/pdf-|lua-|xe-⟩
```

1.2.2 Expansion

\MT@expansion   Set up for expansion?

```
2153 ⟨*pdf-|lua-⟩
2154 \def\MT@expansion{\MT@maybe@do{ex}}
```

\MT@set@ex@codes@s   Setting up font expansion is a bit different because of the selected option. There are two versions of this macro.

  If selected=true, we only apply font expansion to those fonts for which a list has been declared (i.e., like for protrusion).

```
2155 \def\MT@set@ex@codes@s{%
2156   \MT@if@list@exists{%
2157     \MT@get@ex@opt
2158     \let\MT@get@char@unit\relax
2159     \MT@reset@ef@codes
2160     \MT@get@inh@list
2161     \MT@set@inputenc{c}%
2162     \MT@load@list\MT@ex@c@name
2163     \MT@set@listname
2164     \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
2165     \expandafter\MT@set@codes\@tempc,\relax,%
2166     \MT@expandfont
2167   }\relax
2168 }
2169 ⟨/pdf-|lua-⟩
```

\MT@set@ex@codes@n   If, on the other hand, all characters should be expanded by the same amount, we only take the first optional argument to \SetExpansion into account.

\ifMT@nonselected   We need this boolean in \MT@if@list@exists so that no warning for missing lists will be issued.

```
2170 ⟨package⟩\newif\ifMT@nonselected
```

```
2171 ⟨*pdf-|lua-⟩
2172 \def\MT@set@ex@codes@n{%
2173   \MT@nonselectedtrue
2174   \MT@if@list@exists
2175     \MT@get@ex@opt
2176   {%
2177     \let\MT@stretch@  \MT@stretch
2178     \let\MT@shrink@   \MT@shrink
2179     \let\MT@step@     \MT@step
2180     \let\MT@auto@     \MT@auto
2181     \let\MT@ex@factor@\MT@ex@factor
2182   }%
2183   \MT@reset@ef@codes
2184   \MT@expandfont
2185   \MT@nonselectedfalse
2186 }
```

\MT@set@ex@codes   Default is non-selected. It can be changed in the package options.

2187 \let\MT@set@ex@codes\MT@set@ex@codes@n

\MT@expandfont   Expand the font. For some reason, older LuaTeX versions freeze if the autoexpand
modifier is missing. Can't be bothered to find out why. For newer versions, we could
also use the function font.setexpansion, or, in the future, luaotfload's expansion
font feature.

```
2188 ⟨*lua-⟩
2189 \MT@requires@luatex3{
2190 \MT@requires@luatex4{\let\pdffontexpand\expandglyphsinfont}\relax
2191 \ifnum\luatexversion<79
2192 \def\MT@expandfont{%
2193   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ autoexpand\relax
2194 }
2195 \else
2196 \def\MT@expandfont{%
2197   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@\relax
2198 }
2199 \fi
2200 }{
2201 ⟨/lua-⟩
2202 \def\MT@expandfont{%
2203   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
2204 }
2205 ⟨lua-⟩}
```

\MT@set@all@ex   At first, all expansion factors for the characters will be set to 1000 (respectively the
\MT@reset@ef@codes@   factor of this font).

```
2206 \def\MT@set@all@ex#1{%
2207 ⟨debug⟩\MT@dinfo@nl{3}{-- ex: setting all to \number#1}%
2208   \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
2209 }
2210 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}
```

\MT@reset@ef@codes   However, this is only necessary for pdfTeX versions prior to 1.20, or LuaTeX < 0.90
(actually, I think, 0.87).

```
2211 ⟨pdf-⟩\MT@requires@pdftex4
2212 ⟨lua-⟩\MT@requires@luatex5
2213 {
2214   \def\MT@reset@ef@codes{%
2215     \ifnum\MT@ex@factor@=\@m \else
2216       \MT@reset@ef@codes@
2217     \fi
2218   }
2219 }{
2220   \let\MT@reset@ef@codes\MT@reset@ef@codes@
2221 }
```

\MT@ex@split@val   There's only one number per character.

```
2222 \def\MT@ex@split@val#1\relax{%
2223   \@tempcntb=#1\relax
```

Take an optional factor into account.

```
2224   \ifnum\MT@ex@factor@=\@m \else
2225     \MT@scale\@tempcntb \MT@ex@factor@ \@m
2226   \fi
2227   \ifnum\@tempcntb > \MT@ex@max
2228     \MT@warn@ex@too@large\MT@ex@max
2229   \else
2230     \ifnum\@tempcntb < \MT@ex@min
2231       \MT@warn@ex@too@large\MT@ex@min
2232     \fi
2233   \fi
```

```
2234     \efcode\MT@font\MT@char=\@tempcntb
2235 ⟨debug⟩\MT@dinfo@nl{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%
```

Heirs, heirs, I love thy heirs.

```
2236     \MT@ifdefined@c@T\MT@ex@inh@name{%
2237       \MT@ifdefined@n@T{MT@inh@\MT@ex@inh@name @\MT@char @}{%
2238         \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
2239       }%
2240     }%
2241 }
```

\MT@warn@ex@too@large

```
2242 \def\MT@warn@ex@too@large#1{%
2243     \MT@warning@nl{Expansion factor \number\@tempcntb\space too large for
2244       character\MessageBreak `\the\MT@toks' in \MT@curr@list@name.\MessageBreak
2245       Setting it to the maximum of \number#1}%
2246     \@tempcntb=#1\relax
2247 }
```

\MT@get@ex@opt
\MT@ex@factor@
\MT@stretch@
\MT@shrink@
\MT@step@
\MT@auto@

Apply different values to this font?

```
2248 \def\MT@get@ex@opt{%
2249     \MT@set@listname
2250     \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
2251       \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
2252       \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
2253     }{%
2254       \let\MT@ex@factor@\MT@ex@factor
2255     }%
2256     \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
2257     \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
2258     \MT@get@ex@opt@{step}    {Setting expansion step to \number\MT@step@}%
2259 ⟨lua-⟩   \MT@requires@luatex3\relax{%
2260     \MT@get@ex@opt@{auto}{\MT@ifstreq{\MT@auto@}{autoexpand}{En}{Dis}abling automatic expansion}%
2261 ⟨lua-⟩   }%
2262     \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}{%
2263       \MT@preset@ex
2264       \let\MT@reset@ef@codes\relax
2265     }%
2266 }
```

\MT@get@ex@opt@

```
2267 \def\MT@get@ex@opt@#1#2{%
2268     \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
2269       \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
2270       \MT@vinfo{... : #2}%
2271     }{%
2272       \MT@let@nn{MT@#1@}{MT@#1}%
2273     }%
2274 }
```

\MT@set@ex@heirs

```
2275 \def\MT@set@ex@heirs#1{%
2276     \efcode\MT@font#1=\efcode\MT@font\MT@char
2277 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
2278 ⟨debug⟩\MT@dinfo@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
2279 }
```

\MT@preset@ex

```
2280 \def\MT@preset@ex{%
2281     \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
2282     \MT@scale@factor
2283     \MT@set@all@ex\@tempcntb
2284 }
2285 ⟨/pdf-|lua-⟩
```

1.2.3 Interword spacing (glue)

\MT@spacing   Adjustment of interword spacing? Only works with pdfTeX.

```
2286 ⟨*pdf-⟩
2287 \MT@requires@pdftex6{
2288 \def\MT@spacing{\MT@maybe@do{sp}}
```

\MT@set@sp@codes   This is all the same.

```
2289 \def\MT@set@sp@codes{%
2290   \MT@if@list@exists{%
2291     \MT@get@opt
2292     \MT@reset@sp@codes
2293     \MT@get@inh@list
2294     \MT@set@inputenc{c}%
2295     \MT@load@list\MT@sp@c@name
2296     \MT@set@listname
2297     \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
2298     \expandafter\MT@set@codes\@tempc,\relax,%
2299   }\MT@reset@sp@codes
2300 }
```

\MT@sp@split@val   If unit=space, \MT@get@space@unit will be defined to fetch the corresponding
                     fontdimen (2 for the first, 3 for the second and 4 for the third argument).

```
2301 \def\MT@sp@split@val#1,#2,#3\relax{%
2302   \def\@tempb{#1}%
2303   \MT@ifempty\@tempb\relax{%
2304     \MT@get@space@unit2%
2305     \MT@scale@to@em
2306     \knbscode\MT@font\MT@char=\@tempcntb
2307 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
2308   }%
2309   \def\@tempb{#2}%
2310   \MT@ifempty\@tempb\relax{%
2311     \MT@get@space@unit3%
2312     \MT@scale@to@em
2313     \stbscode\MT@font\MT@char=\@tempcntb
2314 ⟨debug⟩\MT@dinfo@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
2315   }%
2316   \def\@tempb{#3}%
2317   \MT@ifempty\@tempb\relax{%
2318     \MT@get@space@unit4%
2319     \MT@scale@to@em
2320     \shbscode\MT@font\MT@char=\@tempcntb
2321 ⟨debug⟩\MT@dinfo@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
2322   }%
2323   \MT@ifdefined@c@T\MT@sp@inh@name{%
2324     \MT@ifdefined@n@T{MT@inh@\MT@sp@inh@name @\MT@char @}{%
2325       \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
2326     }%
2327   }%
2328 }
```

\MT@set@sp@heirs

```
2329 \def\MT@set@sp@heirs#1{%
2330   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
2331   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
2332   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
2333 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
2334 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
2335 ⟨debug⟩          \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
2336 }
```

\MT@set@all@sp
\MT@reset@sp@codes   `2337 \def\MT@set@all@sp#1#2#3{%`
\MT@reset@sp@codes@

```
2338 ⟨debug⟩\MT@dinfo@nl{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
2339    \let\MT@temp\@empty
2340    \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
2341    \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
2342    \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
2343    \MT@do@font\MT@temp
2344 }
2345 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
2346 \let\MT@reset@sp@codes\relax
```

\MT@preset@sp
\MT@preset@sp@
```
2347 \def\MT@preset@sp{%
2348    \expandafter\expandafter\expandafter\MT@preset@sp@
2349      \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
2350 }
2351 \def\MT@preset@sp@#1,#2,#3\@nil{%
2352    \ifx\MT@sp@unit@\@empty
2353      \MT@warn@preset@towidth{sp}%
2354      \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
2355      \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
2356      \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
2357    \else
2358      \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
2359      \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
2360      \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
2361    \fi
2362    \MT@set@all@sp\@tempa\@tempc\@tempb
2363 }
2364 }\relax
```

1.2.4 Additional kerning

\MT@kerning   Again, only check for additional kerning for new versions of pdfTeX.

```
2365 \MT@requires@pdftex6{
2366 \def\MT@kerning{\MT@maybe@do{kn}}
```

\MT@set@kn@codes   It's getting boring, I know.

```
2367 \def\MT@set@kn@codes{%
2368    \MT@if@list@exists{%
2369      \MT@get@opt
2370      \MT@reset@kn@codes
2371      \MT@get@inh@list
2372      \MT@set@inputenc{c}%
2373      \MT@load@list\MT@kn@c@name
2374      \MT@set@listname
2375      \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
2376      \expandafter\MT@set@codes\@tempc,\relax,%
2377    }\MT@reset@kn@codes
2378 }
```

\MT@kn@split@val   Again, the unit may be measured in the space dimension; this time only \fontdimen 2.

```
2379 \def\MT@kn@split@val#1,#2\relax{%
2380    \def\@tempb{#1}%
2381    \MT@ifempty\@tempb\relax{%
2382      \MT@get@space@unit2%
2383      \MT@scale@to@em
2384      \knbccode\MT@font\MT@char=\@tempcntb
2385 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
2386    }%
2387    \def\@tempb{#2}%
2388    \MT@ifempty\@tempb\relax{%
2389      \MT@get@space@unit2%
2390      \MT@scale@to@em
2391      \knaccode\MT@font\MT@char=\@tempcntb
```

```
2392 ⟨debug⟩\MT@dinfo@nl{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
2393     }%
2394     \MT@ifdefined@c@T\MT@kn@inh@name{%
2395       \MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{%
2396         \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
2397       }%
2398     }%
2399 }
```

\MT@set@kn@heirs

```
2400 \def\MT@set@kn@heirs#1{%
2401   \knbccode\MT@font#1=\knbccode\MT@font\MT@char
2402   \knaccode\MT@font#1=\knaccode\MT@font\MT@char
2403 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
2404 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
2405 ⟨debug⟩                              \number\knaccode\MT@font\MT@char}%
2406 }
```

\MT@set@all@kn

\MT@reset@kn@codes
\MT@reset@kn@codes@

```
2407 \def\MT@set@all@kn#1#2{%
2408 ⟨debug⟩\MT@dinfo@nl{3}{-- knac/knbc: setting all to #1/#2}%
2409   \let\MT@temp\@empty
2410   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbccode\MT@font\@tempcnta=#1\relax}}%
2411   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
2412   \MT@do@font\MT@temp
2413 }
2414 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
2415 \let\MT@reset@kn@codes\relax
```

\MT@preset@kn

\MT@preset@kn@

```
2416 \def\MT@preset@kn{%
2417   \expandafter\expandafter\expandafter\MT@preset@kn@
2418     \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
2419 }
2420 \def\MT@preset@kn@#1,#2\@nil{%
2421   \ifx\MT@kn@unit@\@empty
2422     \MT@warn@preset@towidth{kn}%
2423     \let\MT@preset@aux\MT@preset@aux@factor
2424   \else
2425     \def\MT@preset@aux{\MT@preset@aux@space2}%
2426   \fi
2427   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
2428   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
2429   \MT@set@all@kn\@tempa\@tempb
2430 }
2431 }\relax
2432 ⟨/pdf-⟩
```

1.2.5   Tracking

This only works with pdfTEX 1.40 or LuaTEX 0.62.

```
2433 ⟨*pdf-|lua-⟩
2434 ⟨pdf-⟩\MT@requires@pdftex6
2435 ⟨lua-⟩\MT@requires@luatex3
2436 {
```

\MT@tracking
\MT@tracking@
\MT@tr@font@list

We only check whether a font should not be letterspaced at all, not whether we've already done that (because we have to do it again).

```
2437 \let\MT@tr@font@list\@empty
2438 \def\MT@tracking@{%
2439   \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
2440   \ifMT@inlist@\else
2441     \MT@maybe@do{tr}%
```

```
2442      \ifMT@do\else
2443        \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
2444      \fi
2445    \fi
2446 }
2447 ⟨/pdf-|lua-⟩
2448 ⟨pdf-|lua-|letterspace⟩\let\MT@tracking
2449 ⟨pdf-|lua-⟩    \MT@tracking@
2450 ⟨letterspace⟩   \relax
```

\MT@set@tr@codes The tracking amount is determined by the optional argument to \textls, settings from \SetTracking, or the global letterspace option, in this order.

Tracking won't work with older pdfTeX versions ($< 1.40.23$) if the original font's \fontdimen 6 is zero, in which case we issue a warning (once for every font).

```
2451 ⟨*pdf-|lua-|letterspace⟩
2452 \def\MT@set@tr@codes{%
2453 ⟨*pdf-|lua-⟩
2454    \MT@vinfo{Tracking font `\MT@@font'\on@line}%
2455 ⟨*pdf-⟩
2456    \MT@requires@pdftex8\@firstofone{%
2457      \MT@ifdefined@n@TF{\MT@@font-fake6}{%
2458        \expandafter\ifx\csname\MT@@font-fake6\endcsname\@empty
2459          \MT@warning@nl{%
2460            Font `\MT@@font' does not specify its\MessageBreak
2461            \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
2462            tracking will not work with this font}%
2463          \MT@glet@nc{\MT@@font-fake6}\relax
2464        \fi
2465      }%
2466    }{%
2467 ⟨/pdf-⟩
2468    \MT@if@list@exists
2469      \MT@get@tr@opt
2470      \relax
2471 ⟨/pdf-|lua-⟩
2472    \MT@ifdefined@c@TF\MT@letterspace@\relax{\let\MT@letterspace@\MT@letterspace}%
2473    \ifnum\MT@letterspace@=\z@
```

Zero tracking requires special treatment.

```
2474      \MT@set@tr@zero
2475    \else
2476 ⟨pdf-|lua-⟩    \MT@vinfo{... Tracking by \number\MT@letterspace@}%
```

Letterspacing only works in PDF mode.

```
2477      \MT@warn@tracking@DVI
```

\MT@lsfont The letterspaced font instances are saved in macros \⟨*font name*⟩/⟨*letterspacing amount*⟩ls.

In contrast to \MT@font, which may reflect the font characteristics more accurately (taking substitutions into account), \font@name is guaranteed to correspond to an actual font identifier.

```
2478      \xdef\MT@lsfont{\csname\expandafter\string\font@name
2479                        /\number\MT@letterspace@ ls\endcsname}%
2480      \expandafter\ifx\MT@lsfont\relax
2481 ⟨debug⟩\MT@dinfo@nl{1}{... new letterspacing instance}%
```

In case of nested letterspacing with different amounts, we have to extract the base font again.

```
2482        \MT@get@ls@basefont
```

luaotfload provides the faux font feature kernfactor, which we will use when dealing with non-legacy fonts, as it is less problematic and faster than the pdfTeX

primitive \letterspacefont.

```
2483 ⟨*lua-|letterspace⟩
2484      \MT@if@luaotf@font{%
2485 ⟨lua-&debug⟩\MT@dinfo@nl{1}{... luaotf font: \MessageBreak
2486 ⟨lua-&debug⟩          \expandafter\fontname\font@name}%
2487        \global\expandafter\font\MT@lsfont=\MT@ls@fontspec@font
2488      }{%
2489 ⟨/lua-|letterspace⟩
2490 ⟨lua-&debug⟩\MT@dinfo@nl{1}{... legacy font}%
2491        \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
2492 ⟨lua-|letterspace⟩      }%
```

Scale interword spacing (not configurable in letterspace).

```
2493 ⟨*pdf-|lua-⟩
2494      \MT@ifdefined@c@TF\MT@tr@ispace
2495        {\let\@tempa\MT@tr@ispace}%
2496        {\edef\@tempa{\MT@letterspace@*,,}}%
2497      \MT@ifdefined@c@TF\MT@tr@ospace
2498        {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
2499        {\edef\@tempa{\@tempa,,,}}%
2500      \expandafter\MT@tr@set@space\@tempa,%
2501 ⟨/pdf-|lua-⟩
2502 ⟨*letterspace⟩
2503      % spacing = {<letterspace amount>*,,}
2504      \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
2505                                        * \fontdimen2\MT@lsfont/1000\relax
2506 ⟨/letterspace⟩
```

Adjust outer kerning (microtype only).

```
2507 ⟨*pdf-|lua-⟩
2508      \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
2509      \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in letterspace).

```
2510      \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
2511 ⟨/pdf-|lua-⟩
2512 ⟨*letterspace⟩
2513      % no ligatures = {f}
2514      \tagcode\MT@lsfont`f=\m@ne
2515 ⟨/letterspace⟩
```

Adjust protrusion values now, and maybe later (in \MT@pr@split@val) (not for LuaTeX, though, where letterspacing does not interfere with protrusion).

```
2516 ⟨lua-|letterspace⟩      \MT@if@luaotf@font\relax{%
2517 ⟨debug⟩\MT@dinfo@nl{2}{... compensating for tracking (\number\MT@letterspace@)}%
2518      \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
2519                 \rpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
2520      \let\MT@the@pr@code\MT@the@pr@code@tr
2521 ⟨lua-|letterspace⟩      }%
2522      \fi
```

Finally, let the letterspaced font propagate. With LuaTeX, we also need to load.

```
2523      \aftergroup\MT@set@lsfont
2524 ⟨pdf-|lua-⟩      \let\MT@font\MT@lsfont
2525 ⟨lua-⟩      \MT@if@luaotf@font\MT@font\relax
```

\MT@set@curr@ls   We need to remember the current letterspacing amount (for \lslig).

\MT@curr@ls
```
2526      \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
2527      \aftergroup\MT@set@curr@ls
```

Adjust surrounding spacing and kerning.

\MT@set@curr@os   We get the current outer spacing and adjust it, then, after the end of the current outer group, set the current outer spacing, again, and adjust.

```
2528 ⟨*pdf-|lua-⟩
```

```
2529        \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
2530        \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
2531        \MT@tr@outer@l
2532 ⟨/pdf-|lua-⟩
```

If `\MT@ls@adjust` is empty, it's the starred version of `\textls`. Use scaling to avoid a 'Dimension too large'.

```
2533        \ifx\MT@ls@adjust\@empty
2534 ⟨letterspace⟩       % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
2535          \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
2536          \MT@ls@outer@k
```

Otherwise, get the current outer kerning and adjust it, for left and right side (`microtype` only).

```
2537 ⟨*pdf-|lua-⟩
2538        \else
2539          \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
2540                       \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
2541          \ifdim\MT@outer@kern=\z@\else \MT@ls@outer@k \fi
2542          \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
2543                       \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
2544 ⟨/pdf-|lua-⟩
2545 ⟨*letterspace⟩
2546          \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
2547          \MT@afteraftergroup{%
2548            \MT@set@curr@ok
2549            \noexpand\MT@ls@outer@k
2550          }%
2551 ⟨/letterspace⟩
2552        \fi
2553 ⟨*pdf-|lua-⟩
```

`\MT@set@curr@ok` Carry the outer kerning amount to outside the next group, then set outer spacing (which will set kerning, if no space follows).

```
2554        \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
```

Stuff to be done after the letterspace group. The `letterspace` package only adjusts the kerning.

```
2555        \MT@afteraftergroup{%
2556          \MT@set@curr@os
2557          \MT@set@curr@ok
2558          \noexpand\MT@tr@outer@r
2559        }%
2560 ⟨/pdf-|lua-⟩
2561      \fi
2562 ⟨pdf-⟩   }%
2563 }
```

`\MT@afteraftergroup` This helper macro carries stuff outside of the current group to the end of the next group, but will then respect grouping, which is crucial for nested letterspacing. (Following an idea of Will Robertson.)

```
2564 \def\MT@afteraftergroup#1{%
2565 ⟨!letterspace⟩  \MT@maybe@gobble@with@tikz{%
2566      \MT@ifdefined@n@TF{MT@aftergroup@\number\currentgrouplevel}\relax{%
2567        \MT@exp@cs\xdef{MT@aftergroup@\number\currentgrouplevel}%
2568          {\MT@exp@cs\MT@glet{MT@aftergroup@\number\currentgrouplevel}\noexpand\@undefined#1}%
2569        \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
2570          {MT@aftergroup@\number\currentgrouplevel}%
2571      }%
2572 ⟨!letterspace⟩  }%
2573 }
2574 ⟨/pdf-|lua-|letterspace⟩
```

\MT@ls@fontspec@font   Add the kernfactor feature to a font loaded by fontspec.

```
2575 ⟨*lua-|letterspace⟩
2576 \def\MT@ls@fontspec@font{%
2577   \MT@lua{microtype.add_ls([[\MT@letterspace@]])}%
2578 }
2579 ⟨/lua-|letterspace⟩
2580 ⟨*luafile⟩
2581 local function add_ls(k)
2582   local f = tex.fontname(font.current())
2583   local spec,size = match(f,'^(.+)( at .+)$')
2584   if not spec then spec = f end
2585   local a,b,c = match(spec,'^([^:]+):?([^:]*):?(.*)$')
2586   local ls = "kernfactor=" .. k/1000 .. ';'
2587   microtype.sprint(a..':')
2588   if (a == "name" or a == "file") then
2589     microtype.sprint(b..':'..ls..c)
2590   else
2591     microtype.sprint(ls..b)
2592   end
2593   if size then
2594     microtype.sprint(size)
2595   end
2596 end
2597 microtype.add_ls = add_ls
2598
2599 ⟨/luafile⟩
```

\MT@get@tr@opt   Various settings (only for the microtype version).

```
2600 ⟨*pdf-|lua-⟩
2601 \def\MT@get@tr@opt{%
2602   \MT@set@listname
2603   \let\MT@tr@factor@\@m
```

\MT@tr@unit@   Different unit (for letterspace and/or (outer)spacing)?

```
2604   \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @unit}{%
2605     \MT@let@cn\MT@tr@unit@{MT@tr@c@\MT@tr@c@name @unit}%
2606     \ifdim\MT@tr@unit@=1em
2607       \let\MT@tr@unit@\@undefined
2608     \else
2609       \MT@get@unit\MT@tr@unit@
2610     \fi
2611   }%
2612   \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name}{%
2613     \MT@let@cn\MT@letterspace{MT@tr@c@\MT@tr@c@name}%
2614     \MT@ifdefined@c@T\MT@tr@unit@{%
2615       \let\@tempb\MT@letterspace
2616       \MT@scale@to@em
2617       \edef\MT@letterspace{\number\@tempcntb}%
2618     }%
2619   }%
```

\MT@tr@ispace   Adjust interword spacing.
\MT@tr@ospace

```
2620   \MT@get@tr@opt@{spacing}      {ispace}%
2621   \MT@get@tr@opt@{outerspacing}{ospace}%
```

\MT@tr@okern   Adjust outer kerning.

```
2622   \MT@get@tr@opt@{outerkerning}{okern}%
```

\MT@tr@ligatures   Which ligatures should we disable (empty means all, undefined none)?

```
2623   \MT@get@tr@opt@{noligatures} {ligatures}%
2624 }
```

\MT@get@tr@opt@

```
2625 \def\MT@get@tr@opt@#1#2{%
```

```
2626    \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @#1}%
2627      {\MT@let@nn{MT@tr@#2}{MT@tr@c@\MT@tr@c@name @#1}}}%
2628 }
2629 ⟨/pdf-|lua-⟩
```

\MT@set@lsfont   Redefine \font@name, which will be called a second later (in \selectfont).

```
2630 ⟨*pdf-|lua-|letterspace⟩
2631 ⟨plain⟩\MT@requires@latex2{
2632 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}
```

\lsstyle   Disable the tests whether the font should be letterspaced, then trigger the setup.
Only \textls can be used in math mode (\lsstyle may be used inside another
text switch, of course). Still, we have to ensure that math fonts are set up again.
Setting \glb@currsize globally to \@empty (our previous solution) could throw us
into an infinite loop (e.g., with the psnfss packages, via \every@math@size), so
we issue \glb@settings instead. However, in certain situations, we may still miss
some math fonts, so let's try to also enforce it by emptying \glb@currsize, fingers
crossed. The overhead seems small.

```
2633 \DeclareRobustCommand\lsstyle{%
2634    \not@math@alphabet\lsstyle\textls
2635    \let\glb@currsize\@empty
2636 ⟨pdf-|lua-⟩  \MT@maybe@gobble@with@tikz{\aftergroup\glb@settings}%
2637 ⟨pdf-|lua-⟩  \def\MT@feat{tr}%
2638    \let\MT@tracking\MT@set@tr@codes
2639    \selectfont
2640 }
```

Now the definitions for the letterspace package with plain TEX.

```
2641 ⟨*plain⟩
2642 }{
2643 \def\MT@set@lsfont{\MT@lsfont}
2644 \def\lsstyle{%
2645    \begingroup
2646    \escapechar\m@ne
2647    \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
2648    \MT@set@tr@codes
2649    \endgroup
2650 }
2651 \let\textls\@undefined
2652 \let\lslig\@undefined
2653 }
2654 ⟨/plain⟩
```

\lslig   For Fraktur fonts, some ligatures shouldn't be broken up. This command will
temporarily select the base font and insert the correct kerning.

```
2655 \DeclareRobustCommand\lslig[1]{%
2656    {\MT@ifdefined@c@TF\MT@curr@ls{%
2657       \escapechar\m@ne
2658       \MT@get@ls@basefont
2659       \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
2660       \kern\MT@outer@kern
2661       \font@name #1%
2662       \kern\MT@outer@kern
2663    }{#1}}%
2664 }
```

\MT@ls@basefont   pdfTEX cannot letterspace fonts that already are letterspaced. Therefore, we have
\MT@get@ls@basefont   to save the base font in \⟨font name⟩@base.
                        The previous solution (checking the macro's meaning with \pdfmatch), where
we were loading the base font via the \font primitive again, would destroy all
previously set up micro-typographic features of the font.

```
2665 \def\MT@get@ls@basefont{%
2666   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2667   \expandafter\ifx\MT@ls@basefont\relax
2668     \MT@exp@two@c\MT@glet\MT@ls@basefont\font@name
2669   \else
2670 ⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
2671     \MT@exp@two@c\let\font@name\MT@ls@basefont
2672   \fi
2673 }
```

\MT@set@lsbasefont   If tracking is switched off in the middle of the document, or if \textls is called
\MT@set@tr@zero   with a zero letterspacing amount, we have to retrieve the base font and select it.

```
2674 \def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
2675 \def\MT@set@tr@zero{%
2676 ⟨debug⟩\MT@dinfo@nl{1}{... zero tracking}%
2677   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2678   \expandafter\ifx\MT@ls@basefont\relax \else
2679 ⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
2680     \aftergroup\MT@set@lsbasefont
2681   \fi
2682 }
2683 ⟨/pdf-|lua-|letterspace⟩
```

\MT@tr@noligatures   pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

```
2684 ⟨*pdf-|lua-⟩
2685 ⟨pdf-⟩\MT@requires@pdftex7{
2686   \def\MT@tr@noligatures{%
2687     \ifx\MT@tr@ligatures\@empty
2688       \MT@noligatures@\MT@lsfont\@undefined
2689     \else
2690       \MT@noligatures@\MT@lsfont\MT@tr@ligatures
2691     \fi
2692   }
2693 ⟨*pdf-⟩
2694 }{
2695   \def\MT@tr@noligatures{%
2696     \MT@warning@nl{%
2697       Disabling selected ligatures is only possible since\MessageBreak
2698       pdftex 1.40.4. Disabling all ligatures instead}%
2699     \MT@glet\MT@tr@noligatures\relax
2700   }
2701 }
2702 ⟨/pdf-⟩
```

\MT@outer@space   A new skip for outer spacing.

```
2703 \newskip\MT@outer@space
```

\MT@tr@set@space   Adjust interword spacing (\fontdimen 2,3,4) for inner and outer space. For inner
                   spacing, the font dimensions will be adjusted, the settings for outer spacing will be
                   remembered in a macro.

```
2704 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
2705 ⟨debug⟩\MT@dinfo@nl2{... orig. space: \the\fontdimen2\MT@lsfont,
2706 ⟨debug⟩     \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
2707 ⟨debug⟩     \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
2708   \let\MT@temp\@empty
2709   \MT@tr@set@space@{#1}{#4}{2}\@empty
2710   \MT@tr@set@space@{#2}{#5}{3}\@plus
2711   \MT@tr@set@space@{#3}{#6}{4}\@minus
2712   \MT@glet@nc{MT@outer@space\expandafter\string\font@name}\MT@temp
2713 ⟨debug⟩\MT@dinfo@nl2{... inner space: \the\fontdimen2\MT@lsfont,
2714 ⟨debug⟩     \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
2715 ⟨debug⟩\MT@dinfo@nl2{... outer space: \MT@temp}%
2716 }
```

\MT@tr@set@space@   If settings for outer spacing ⟨*#2*⟩ don't exist, they will be inherited from the inner
spacing settings ⟨*#1*⟩.

```
2717 \def\MT@tr@set@space@#1#2#3#4{%
2718   \MT@ifempty{#2}{%
2719     \MT@ifempty{#1}\relax{%
2720       \MT@tr@set@space@@{#1}{#3}{1000}%
2721       \fontdimen#3\MT@lsfont=\@tempdima
2722     }%
2723     \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
2724   }{%
2725     \MT@tr@set@space@@{#2}{#3}{2000}%
2726     \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2727     \MT@ifempty{#1}\relax{%
2728       \MT@tr@set@space@@{#1}{#3}{1000}%
2729       \fontdimen#3\MT@lsfont=\@tempdima
2730     }%
2731   }%
2732 }
```

\MT@tr@set@space@@   If the value is followed by an asterisk, the fontdimen will be scaled by the respective
amount, otherwise the value denotes the desired dimension in the respective unit.

```
2733 \def\MT@tr@set@space@@#1#2#3{%
2734   \MT@test@ast#1*\@nil{%
2735     \MT@ifdefined@c@TF\MT@tr@unit@
2736       {\edef\@tempb{#1}\MT@scale@to@em}
2737       {\@tempcntb=#1\relax}%
2738     \@tempdima=\dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
```

For \fontdimen 2, we also have to subtract the kerning that letterspacing adds to
each side of the characters (only half if it's for outer spacing).

```
2739     \ifnum#2=\tw@
2740       \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
2741     \fi
2742   }{%
2743     \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
2744     \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
2745   }%
2746 ⟨debug⟩\MT@dinfo@nl3{... : font dimen #2 (#1): \the\@tempdima}%
2747 }
```

\MT@tr@outer@l   Recall the last skip (must really be an interword space, not just a marker, nor a
'hard' space, i.e., one that doesn't contain stretch or shrink parts).

```
2748 \def\MT@tr@outer@l{%
2749   \ifhmode
2750     \ifdim\lastskip>5sp
2751       \edef\x{\the\lastskip minus 0pt}%
2752       \setbox\z@\hbox{\MT@outer@space=\x}%
2753       \ifdim\wd\z@>\z@
2754 ⟨debug⟩\MT@dinfo2{[[[ adjusting pre space: \the\MT@outer@space}%
2755         \unskip \hskip\MT@outer@space\relax
```

Disable left outer kerning.

```
2756         \let\MT@ls@outer@k\relax
2757       \else
```

The ragged2e package sets \spaceskip without glue.

```
2758         \ifdim\lastskip=%
2759           \ifnum\spacefactor<2000
2760             \spaceskip
2761           \else
2762             \ifdim\xspaceskip=\z@
2763               \dimexpr\spaceskip+\fontdimen7\font@name\relax
2764             \else
2765               \xspaceskip
```

```
2766                    \fi
2767                  \fi
2768 ⟨debug⟩\MT@dinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
2769                \unskip \hskip\MT@outer@space\relax
2770                \let\MT@ls@outer@k\relax
2771              \fi
2772            \fi
2773         \fi
2774     \fi
2775 }
```

\MT@tr@outer@next microtype also adjusts spacing. The following is borrowed from soul. I've added the
\MT@tr@outer@r cases for italic correction, since tracking may also be triggered by text commands
(e.g., \textsc).

```
2776 \def\MT@tr@outer@r{%
2777   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
2778 }
```

\MT@if@outer@next We avoid using \ifx tests, in case \MT@tr@outer@next is \let to \fi etc.

```
2779 \def\MT@if@outer@next#1{%
2780   \ifx\MT@tr@outer@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2781 }
```

\MT@tr@outer@r@

```
2782 \def\MT@tr@outer@r@{%
2783   \def\MT@temp*{}%
```

Don't adjust in math mode. There was a tricky bug when \textls was the last
command in a \mathchoice group.

```
2784   \ifmmode \else
```

A similar bug occurred when adjustment would happen inside a discretionary
group, which we prevent here. This only works with e-TeX (which we know is
available).

```
2785     \ifnum\currentgrouptype=10 \else
2786       \def\MT@temp*##1{\ifmode\hskip\MT@outer@space
2787 ⟨debug⟩\MT@dinfo2{]]] adjusting post space (1): \the\MT@outer@space}%
2788       \fi}%
2789     \expandafter\ifcat\expandafter\noexpand\csname MT@tr@outer@next\endcsname\egroup

2790       \ifhmode\unkern\fi\egroup
2791       \MT@set@curr@ok \MT@set@curr@os
2792       \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
2793     \else
```

If the next token is \maybe@ic (from an enclosing text command), we gobble it,
read the next one, feed it to \maybe@ic@ (via \MT@tr@outer@icr) and then call
ourselves again.

```
2794       \MT@if@outer@next\maybe@ic{%
2795         \MT@set@curr@ok \MT@set@curr@os
2796         \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}%
2797       }{%
```

If the next token is \check@icr (from an inner text command), we insert ourselves
just before it. This will then call \maybe@ic again the next round (which however
will always insert an italic correction, since it doesn't read beyond our group).

```
2798         \MT@if@outer@next\check@icr{%
2799           \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}%
2800         }{%
2801           \MT@if@outer@next\@sptoken{%
2802             \def\MT@temp* {\ifmode\hskip\MT@outer@space
2803 ⟨debug⟩\MT@dinfo2{]]] adjusting post space (2): \the\MT@outer@space}%
```

```
2804                    \fi}%
2805                }{%
2806               \MT@if@outer@next~{%
2807                 \def\MT@temp*~{\nobreak\hskip\MT@outer@space
2808 ⟨debug⟩\MT@dinfo2{]]] adjusting post space (3): \the\MT@outer@space}%
2809                   }%
2810                }{%
2811                 \MT@if@outer@next\ \relax{%
2812                   \MT@if@outer@next\space\relax{%
2813                     \MT@if@outer@next\@xobeysp\relax{%
```

xspace requires special treatment.

```
2814                       \MT@if@outer@next\xspace{%
2815                         \def\MT@temp*\xspace{\MT@xspace}%
2816                       }{%
```

If there's no outer spacing, there may be outer kerning.

```
2817                         \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k
2818 ⟨debug⟩\MT@dinfo2{--- adjusting post kern: \the\MT@outer@kern}%
2819                           \fi}%
2820                         \MT@let@nc{MT@tr@outer@next}\relax
2821      }}}}}}}}\fi
2822    \fi\fi
2823    \MT@temp*%
2824 }
```

\MT@tr@outer@icr   Helper macros for the italic correction mess.

\MT@tr@outer@icr@
```
2825 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
2826 \def\MT@tr@outer@icr@{%
2827   \let\@let@token= \MT@tr@outer@next
2828   \maybe@ic@
2829 }
```

\MT@xspace   If the group is followed by \xspace, we first feed \xspace with the next token, then
\MT@xspace@   check whether it has inserted a space. \@let@token might be something evil, so it
                        should be encapsulated here.

```
2830 \def\MT@xspace{\futurelet\@let@token\MT@xspace@}
2831 \def\MT@xspace@{\@xspace@firsttrue\@xspace
2832   \ifdim\lastskip>5sp
2833     \unskip \hskip\MT@outer@space
2834   \else
2835     \ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k \fi
2836   \fi
2837 }
```

For older pdfTeX versions and LuaTeX, throw an error.

```
2838 }{
2839   \DeclareRobustCommand\lsstyle{%
2840     \MT@error{Letterspacing only works with \MT@engine tex version
2841 ⟨pdf-⟩    1.40%
2842 ⟨lua-⟩    0.62%
2843       \MessageBreak or newer}
2844       {Upgrade \MT@engine tex, or try the `soul' package instead.}%
2845     \MT@glet\lsstyle\relax
2846   }
2847 }
```

And for XꟺTEX, too.

```
2848 ⟨/pdf-|lua-⟩
2849 ⟨*xe-⟩
2850 \DeclareRobustCommand\lsstyle{%
2851   \MT@error{Letterspacing currently doesn't work with xetex}
2852         {Run pdftex or luatex, or use the `soul' package instead.}%
2853   \MT@glet\lsstyle\relax
2854 }
```

2855 ⟨/xe-⟩

\textls
\MT@ls@adjust@

This command may be used like the other text commands. The starred version removes kerning on the sides. The optional argument changes the letterspacing factor.

```
2856 ⟨*package | letterspace⟩
2857 \DeclareRobustCommand\textls{%
2858   \@ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%
2859           {\let\MT@ls@adjust@\MT@ls@adjust@relax\MT@textls}%
2860 }
```

\MT@textls
\MT@letterspace@

This is now almost LATEX's \DeclareTextFontCommand, with the difference that we adjust the outer spacing and kerning also for \lsstyle, while LATEX's text *switches* don't bother about italic correction.

```
2861 \newcommand\MT@textls[2][]{%
2862   \ifmmode
2863     \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
2864   \else
2865     \hmode@bgroup
2866       \MT@ls@set@ls{#1}%
2867       \lsstyle #2%
2868       \expandafter
2869     \egroup
2870   \fi
2871 }
```

\MT@ls@adjust
\MT@ls@adjust@empty
\MT@ls@adjust@relax
\MT@ls@set@ls

Set current letterspacing amount and outer kerning. This has to be done inside the same group as the letterspacing command.

```
2872 \def\MT@ls@adjust@empty{\let\MT@ls@adjust\@empty}
2873 \def\MT@ls@adjust@relax{\let\MT@ls@adjust\relax}
2874 \def\MT@ls@set@ls#1{%
2875   \MT@ifempty{#1}%
2876     {\let\MT@letterspace@\@undefined}%
2877     {\KV@@sp@def\MT@letterspace@{#1}%
2878      \edef\MT@letterspace@{\number\MT@letterspace@}%
2879      \MT@ls@too@large\MT@letterspace@}%
2880   \MT@ls@adjust@
2881 }
```

\MT@ls@too@large

Test whether letterspacing amount is too large.

```
2882 \def\MT@ls@too@large#1{%
2883   \ifnum#1>\MT@tr@max
2884     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
2885     \edef#1{\number\MT@tr@max}%
2886   \else
2887     \ifnum#1<\MT@tr@min
2888       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
2889       \edef#1{\number\MT@tr@min}%
2890     \fi
2891   \fi
2892 }
```

\MT@outer@kern
\MT@tr@set@okern

This dimen is used for the starred version of \textls, for \lslig and for adjusted outer kerning.

```
2893 \newdimen\MT@outer@kern
2894 ⟨/package | letterspace⟩
2895 ⟨*pdf- | lua-⟩
2896 \def\MT@tr@set@okern#1,#2,{%
2897   \let\MT@temp\@empty
2898   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
2899   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
2900   \MT@glet@nc{MT@outer@kern\expandafter\string\font@name}\MT@temp
2901 ⟨debug⟩\MT@dinfo@nl2{... outer kerning: (#1,#2)
```

```
2902 ⟨debug⟩                      = \@nameuse{MT@outer@kern\expandafter\string\font@name}}%
2903 }
```

\MT@tr@set@okern@

```
2904 \def\MT@tr@set@okern@#1{%
2905   \MT@test@ast#1*\@nil{%
2906     \MT@ifdefined@c@TF\MT@tr@unit@
2907       {\edef\@tempb{#1}\MT@scale@to@em}
2908       {\@tempcntb=#1\relax}%
2909     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
2910   }{%
2911     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
2912     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
2913                       * \fontdimen6\MT@lsfont/2000\relax
2914   }%
2915   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
2916                           * \fontdimen6\MT@lsfont/2000\relax
2917   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
2918 }
2919 ⟨/pdf-|lua-⟩
```

\MT@ls@outer@k   Adjust outer kerning. We additionally add a marker (\kern3sp\kern-3sp) for cases of nested letterspacing without anything actually printed.

```
2920 ⟨*pdf-|lua-|letterspace⟩
2921 \def\MT@ls@outer@k{%
2922   \ifhmode
2923     \ifdim\lastkern=-3sp \unkern
2924       \ifdim\lastkern=3sp \kern-3sp
2925         \expandafter\expandafter\expandafter\@gobble
2926       \else \unkern
2927         \expandafter\expandafter\expandafter\@firstofone
2928       \fi
2929     \else
2930       \expandafter\@firstofone
2931     \fi
2932     {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
2933   \fi
2934 }
2935 ⟨/pdf-|lua-|letterspace⟩
```

1.2.6 Disabling ligatures

\MT@noligatures   The possibility to disable ligatures is a new features of pdfTEX 1.30, and also works with LuaTEX.

```
2936 ⟨*pdf-|lua-⟩
2937 ⟨pdf-⟩\MT@requires@pdftex5{
2938 \def\MT@noligatures{%
2939   \MT@dotrue
2940   \let\@tempa\MT@nl@setname
2941   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
2942     \MT@ifdefined@n@TF{MT@checklist@##1}%
2943       {\csname MT@checklist@##1\endcsname}%
2944       {\MT@checklist@{##1}}%
2945     {nl}%
2946   }%
2947   \ifMT@do
2948     \MT@noligatures@\MT@font\MT@nl@ligatures
2949   \fi
2950 }
```

\MT@noligatures@   This is also used by \MT@set@tr@codes.

```
2951 ⟨lua-⟩\MT@requires@luatex4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
2952 \def\MT@noligatures@#1#2{%
```

```
2953    \MT@ifdefined@c@TF#2{%
```

Early MiKTₑX versions (before 2.5.2579) didn't know \tagcode.

```
2954      \MT@ifdefined@c@TF\tagcode{%
```

No 'inputenc' key.

```
2955        \let\MT@warn@maybe@inputenc\@empty
2956        \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
2957        \MT@map@clist@c#2{%
2958          \KV@@sp@def\@tempa{##1}\MT@get@slot
2959          \ifnum\MT@char>\m@ne
2960            \tagcode#1\MT@char=\m@ne
```

With LuaTₑX, we additionally register the ligatures that should be inhibited in a table (used by the luaotfload function keepligature).

```
2961 ⟨lua-⟩          \MT@if@luaotf@font
2962 ⟨lua-⟩              {\MT@lua{microtype.noligatures([[#1]],[[\MT@char]])}}\relax
2963          \fi
2964        }%
2965        \MT@vinfo{... Disabling ligatures for characters: #2}%
2966      }{%
2967        \pdfnoligatures#1%
2968        \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
2969            know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
2970            the font instead}%
2971      }%
2972    }{%
2973      \pdfnoligatures#1%
2974 ⟨lua-⟩    \MT@if@luaotf@font
2975 ⟨lua-⟩        {\MT@lua{microtype.noligatures([[#1]],"_all_")}}\relax
2976      \MT@vinfo{... Disabling all ligatures}%
2977    }%
2978 }
2979 ⟨pdf-⟩}\relax
2980 ⟨/pdf-|lua-⟩
```

For each potential ligature, luaotfload will call the keepligature function, which expects the first node of the ligature, to check whether they should be kept or inhibited. Here's our concoction of this function. The table microtype.ligs will be populated in \MT@noligatures@.

```
2981 ⟨*luafile⟩
2982 microtype.ligs = microtype.ligs or { }
2983
2984 local function noligatures(fontcs,liga)
2985   local fontcs = match(fontcs,"([^ ]+)")
2986   microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
2987   table.insert(microtype.ligs[fontcs],liga)
2988 end
2989 microtype.noligatures = noligatures
2990
2991 local function keepligature(c)
2992   local nodedirect = node.direct
2993   local getfield  = nodedirect.getfield
2994   local getfont    = nodedirect.getfont
2995   local f,ch
2996   if type(c) == "userdata" then -- in older luaotfload versions, c was a node
2997     f  = c.font
2998     ch = c.components.char
2999   else                          -- since 2.6, c is a (direct node) number
3000     f = getfont(c)
3001     ch = getfield(getfield(c,"components"),"char")
3002   end
3003 -- if ch then -- should always be true
3004   local ligs = microtype.ligs[match(tex.fontidentifier(f),"\\([^ ]+)")]
```

```
3005  if ligs then
3006    for _,lig in pairs(ligs) do
3007      if lig == "_all_" or tonumber(lig) == ch then
3008        return false
3009      end
3010    end
3011  end
3012  return true
3013 -- end
3014 end
3015
3016 if luaotfload and luaotfload.letterspace then
3017   if luaotfload.letterspace.keepligature then
3018     microtype.info("overwriting function `keepligature'")
3019   end
3020   luaotfload.letterspace.keepligature = keepligature
3021 end
3022
3023 ⟨/luafile⟩
```

1.2.7 Loading the configuration

\MT@load@list   Recurse through the lists to be loaded.

```
3024 ⟨*package|show⟩
3025 ⟨package⟩\def\MT@load@list#1%
3026 ⟨show⟩\def\MTS@load@list#1%
3027   {\edef\@tempa{#1}%
3028   \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
3029   \MT@ifstreq\@tempa\@tempb{%
3030     \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempa' cannot load itself}{}%
3031   }{%
3032     \ifx\@tempb\relax
3033 ⟨show⟩      :\par\medskip\leavevmode
3034     \else
3035       \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
3036 ⟨show⟩        \MTS@printtext{, loading \textttt{\@tempb}}%
3037         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `\@tempb'}%
3038         \begingroup
3039           \MT@load@list\@tempb
3040         \endgroup
3041         \edef\MT@curr@list@name{%
3042 ⟨package⟩                  \@nameuse{MT@abbr@\MT@feat} list \noexpand\MessageBreak
3043                   `\@tempb'}%
3044         \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
3045         \expandafter\MT@set@codes\@tempc,\relax,%
3046 ⟨show⟩        \vrule width 4cm height .5pt \\
3047 ⟨show⟩        \MTS@printtext{End of list \textttt{\MT@curr@list@name}}%
3048 ⟨show⟩        \par\medskip\leavevmode
3049       }{%
3050         \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempb' undefined.\MessageBreak
3051                   Cannot load it from list `\@tempa'}{}%
3052       }%
3053     \fi
3054   }%
3055 }
3056 ⟨/package|show⟩
```

\MT@find@file   Micro-typographic settings may be written into a file mt-⟨*font family*⟩.cfg.
\MT@file@list   We must also record whether we've already loaded the file.

```
3057 ⟨*package⟩
3058 \let\MT@file@list\@empty
3059 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```
3060    \MT@in@clist{#1}\MT@file@list
3061    \ifMT@inlist@ \else
```

Don't forget that because reading the files takes place inside a group, all commands that may be used there have to be defined globally.

```
3062      \MT@begin@catcodes
3063        \let\MT@begin@catcodes\relax
3064        \let\MT@end@catcodes\relax
3065        \InputIfFileExists{mt-#1.cfg}{%
3066          \edef\MT@curr@file{mt-#1.cfg}%
3067          \MT@vinfo{... Loading configuration file \MT@curr@file}%
3068          \MT@xadd\MT@file@list{#1,}%
3069        }{%
3070          \MT@get@basefamily#1\@empty\@empty\@empty\@nil
3071          \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
3072          \ifMT@inlist@
3073            \MT@xadd\MT@file@list{#1,}%
3074          \else
3075            \InputIfFileExists{mt-\@tempa.cfg}{%
3076              \edef\MT@curr@file{mt-\@tempa.cfg}%
3077              \MT@vinfo{... Loading configuration file \MT@curr@file}%
3078              \MT@xadd\MT@file@list{\@tempa,#1,}%
3079            }{%
3080              \MT@vinfo{... No configuration file mt-#1.cfg}%
3081              \MT@xadd\MT@file@list{#1,}%
3082            }%
3083          \fi
3084        }%
3085      \endgroup
3086    \fi
3087 }
```

\MT@cfg@catcodes   We have to make sure that all characters have the correct category code. Especially, new lines and spaces should be ignored, since files might be loaded in the middle of the document. This is basically \nfss@catcodes (from the LaTeX kernel). I've added: & (in tabulars), !, ?, ;, : (french), ,, $, _, ~, and = (Turkish babel).

OK, now all printable characters up to 127 are 'other'. We hope that letters are always letters and numbers other. (listings makes them active, see section 1.1.6.)

We leave ^ at catcode 7, so that stuff like '^^ff' remains possible.

```
3088 \def\MT@cfg@catcodes{%
3089   \makeatletter
3090   \catcode`\^7%
3091   \catcode`\ 9%
3092   \catcode`\^^I9%
3093   \catcode`\^^M9%
3094   \catcode`\\\z@
3095   \catcode`\{\@ne
3096   \catcode`\}\tw@
3097   \catcode`\#6%
3098   \catcode`\%14%
3099   \MT@map@tlist@n
3100     {\!\"\$\&\'\(\)\*\+\,\-\.\/\:\;\<\=\>\?\[\]\_\`\|\~}%
3101     \@makeother
3102 }
```

\MT@begin@catcodes   This will be used before reading the files as well as in all configuration commands, so that catcodes are also harmless when these commands are used outside the configuration files.

```
3103 \def\MT@begin@catcodes{%
3104   \begingroup
3105   \MT@cfg@catcodes
```

Table 1:		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Order for matching font attributes	Encoding	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
	Family	•	•	•	•	•	•	•	•	-	-	-	-	-	-	-	-
	Series	•	•	•	•	-	-	-	-	•	•	•	•	-	-	-	-
	Shape	•	•	-	-	•	•	-	-	•	•	-	-	•	•	-	-
	Size	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```
3106 }
```

\MT@end@catcodes End group if outside configuration file (otherwise relax).

```
3107 \let\MT@end@catcodes\endgroup
```

\MT@get@basefamily The family name might have a suffix e.g., for expert set (x), old style numbers (j) swash capitals (w) etc. We mustn't simply remove the last letter, as this would make for instance cms out of cmss *and* cmsy (OK, cmex will still become cme . . .).

 We only work on the font name if it is longer than three characters.

```
3108 \def\MT@get@basefamily#1#2#3#4\@nil{%
3109   \ifx\@empty#4%
3110     \def\@tempa{#1#2#3}%
3111   \else
3112     \let\@tempa\@empty
3113     \edef\@tempb{#1#2#3#4}%
3114     \expandafter\MT@get@basefamily@\@tempb\@nil
3115   \fi
3116 }
```

\MT@get@basefamily@ This will only remove one suffix (the longest match), so that *combinations* of suffixes would have be to added manually (e.g., \DeclareMicrotypeVariants*{aw}). But otherwise, something like 'pplx' would be truncated to 'p'.

```
3117 \def\MT@get@basefamily@#1#2\@nil{%
3118   \edef\@tempa{\@tempa#1}%
3119   \ifx\\#2\\\expandafter\@gobble\else\expandafter\@firstofone\fi
3120   {\MT@in@tlist{#2}\MT@variants
3121    \ifMT@inlist@\else\MT@get@basefamily@#2\@nil\fi}%
3122 }
```

\MT@listname Try all combinations of font family, series, shape and size to get a list for the current font.
\MT@get@listname

\MT@get@listname@
```
3123 \def\MT@get@listname#1{%
3124 ⟨debug⟩\MT@dinfo@nl{1}{trying to find \@nameuse{MT@abbr@#1} list for font `\MT@@font'}%
3125   \let\MT@listname\@undefined
3126   \def\@tempb{#1}%
3127   \MT@map@tlist@c\MT@try@order\MT@get@listname@
3128 }
3129 \def\MT@get@listname@#1{%
3130   \expandafter\MT@next@listname#1%
3131   \ifx\MT@listname\@undefined \else
3132     \expandafter\MT@tlist@break
3133   \fi
3134 }
```

\MT@try@order Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don't need table 1 in the documentation part any longer and can cast it off here.

```
3135 \def\MT@try@order{%
3136   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
3137   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
3138 }
```

\MT@next@listname   The current context is added to the font attributes. That is, the context must match.

```
3139 \def\MT@next@listname#1#2#3#4{%
3140   \ifnum#1=\z@\MT@nofamilytrue\fi
3141   \edef\@tempa{\MT@encoding
3142 /\ifnum#1=\@ne \MT@family  \fi
3143 /\ifnum#2=\@ne \MT@series  \fi
3144 /\ifnum#3=\@ne \MT@shape   \fi
3145 /\ifnum#4=\@ne *\fi
3146              \MT@context}%
3147 ⟨debug⟩\MT@dinfo@nl{1}{trying \@tempa}%
3148   \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
3149     \MT@next@listname@#4%
3150   }{%
```

Also try with an alias family.

```
3151     \ifnum#1=\@ne
3152       \ifx\MT@familyalias\@empty \else
3153         \edef\@tempa{\MT@encoding
3154                     /\MT@familyalias
3155       /\ifnum#2=\@ne \MT@series\fi
3156       /\ifnum#3=\@ne \MT@shape\fi
3157       /\ifnum#4=\@ne *\fi
3158                     \MT@context}%
3159 ⟨debug⟩\MT@dinfo@nl{1}{(alias) \@tempa}%
3160         \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
3161           \MT@next@listname@#4%
3162         }%
3163       \fi
3164     \fi
3165   }%
3166 }
```

\MT@next@listname@   If size is to be evaluated, do that, otherwise use the current list.

```
3167 \def\MT@next@listname@#1{%
3168   \ifnum#1=\@ne
3169     \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
3170     \ifMT@inlist@
3171       \let\MT@listname\MT@size@name
3172     \fi
3173   \else
3174     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
3175   \fi
3176 }
```

\MT@if@list@exists

\MT@context
```
3177 \def\MT@if@list@exists{%
3178   \MT@let@cn\MT@context{MT@\MT@feat @context}%
3179   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
3180   \MT@get@listname{\MT@feat @c}%
3181   \MT@ifdefined@c@TF\MT@listname{%
3182     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
3183     \ifMT@nonselected
3184       \MT@vinfo{... Applying non-selected expansion (list `\MT@listname')}%
3185     \else
3186       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\MT@listname'}%
3187     \fi
3188     \@firstoftwo
3189   }{%
```

Since the name cannot be \@empty, this is a sound proof that no matching list exists.

```
3190     \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```
3191        \ifMT@nonselected
3192          \MT@vinfo{... Applying non-selected expansion (no list)}%
3193        \else
```

Tracking doesn't require a list, either.

```
3194          \MT@ifstreq\MT@feat{tr}\relax{%
3195            \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
3196              for font\MessageBreak`\MT@@font'%
3197                \ifx\MT@context\@empty\else\space(context: `\MT@context')\fi.
3198              Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
3199          }%
3200        \fi
3201        \@secondoftwo
3202    }%
3203 }
```

\MT@get@inh@list   The inheritance lists are global (no context).

\MT@context
```
3204 \def\MT@get@inh@list{%
3205    \let\MT@context\@empty

3206    \MT@get@listname{\MT@feat @inh}%
3207    \MT@ifdefined@c@TF\MT@listname{%
3208      \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
3209 ⟨debug⟩\MT@dinfo@nl{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
3210 ⟨debug⟩                       `\MT@listname'}%
3211      \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%
```

If the list is \@empty, it has already been parsed.

```
3212      \ifx\@tempc\@empty \else
3213 ⟨debug⟩\MT@dinfo@nl{1}{parsing inheritance list ...}%
```

The group is only required in case an input encoding is given.

```
3214        \begingroup
3215        \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
3216        \MT@set@inputenc{inh}%
3217        \expandafter\MT@inh@do\@tempc,\relax,%
3218        \MT@glet@nc{MT@\MT@feat @inh@\MT@listname}\@empty
3219        \endgroup
3220      \fi
3221    }{%
3222      \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
3223    }%
3224 }
```

1.2.8   Translating characters into slots

Get the slot number of the character in the current encoding.

\MT@get@slot   There are lots of possibilities how a character may be specified in the configuration files, which makes translating them into slot numbers quite expensive. Also, we want to have this as robust as possible, so that the user does not have to solve a sphinx's riddle if anything goes wrong.

\MT@char   The character is in \@tempa, we want its slot number in \MT@char.

\MT@char@
```
3225 \def\MT@get@slot{%
3226    \escapechar`\\
3227    \let\MT@char@\m@ne
3228    \MT@noresttrue
```

Save unexpanded string in case we need to issue a warning message.

```
3229    \MT@toks=\expandafter{\@tempa}%
```

It might be an active character, i.e., an 8-bit character defined by inputenc. If so, we will expand it here to its LICR form.

```
3230    \MT@exp@two@c\MT@is@active\string\@tempa\@nil
```

Now, let's walk through (hopefully) all possible cases.

- It's a letter, a character or a number.

```
3231    \expandafter\MT@is@letter\@tempa\relax\relax
3232    \ifnum\MT@char@ < \z@
```

- OK, so it must be a macro. We do not allow random commands but only those defined in LaTeX's idiosyncratic font encoding scheme:

  If \⟨encoding⟩\⟨command⟩ (that's *one* command) is defined, we try to extract the slot number.

  We must be cautious not to stumble over accented characters consisting of two commands, like \'\i or \U\CYRI, hence, \string wouldn't be safe enough.

```
3233      \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
3234        \MT@is@symbol
```

- Now, we'll catch the rest, which hopefully is an accented character (e.g. \"a).

```
3235        {\expandafter\MT@is@composite\@tempa\relax\relax}%
3236      \ifnum\MT@char@ < \z@
```

- It could also be a \chardefed command (e.g., the percent character). This seems the least likely case, so it's last.

```
3237        \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
3238          \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
3239      \fi
3240    \fi
3241    \let\MT@char\MT@char@
3242    \MT@get@slot@
3243    \escapechar\m@ne
3244  }
3245  ⟨/package⟩
```

\MT@get@slot@

```
3246  ⟨*pdf-|lua-|xe-⟩
3247  \def\MT@get@slot@{%
```

If it's a legacy (i.e., TFM) font, proceed as usual.

```
3248  ⟨xe-⟩  \ifnum\XeTeXfonttype\MT@font=\z@
3249    \ifnum\MT@char > \m@ne
```

In LuaTeX, it may also be a glyph name, prefixed with '/'.

```
3250  ⟨*lua-⟩
3251      \ifnum\MT@char=47\relax
3252        \ifMT@norest \else
3253          \@tempcnta=\MT@lua{
3254            local glyph = microtype.name_to_slot([[\expandafter\@gobble\@tempa]],true)
3255            if glyph then tex.write(glyph)
3256            else tex.write(-1)
3257            end
3258          }\relax
3259          \ifnum\@tempcnta<\z@
3260            \MT@warn@unknown
3261            \let\MT@char\m@ne
3262          \else
3263            \edef\MT@char{\the\@tempcnta}%
```

```
3264 ⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}%
3265         \fi
3266       \fi
3267     \else
3268 ⟨/lua-⟩
```

If the user has specified something like 'fi', or wanted to define a number but forgot to use three digits, we'll have something left of the string. In this case, we issue a warning and forget the complete string.

```
3269     \ifMT@norest \else
3270       \MT@warn@rest
3271 ⟨pdf-|lua-⟩        \let\MT@char\m@ne
3272 ⟨xe-⟩        \let\MT@char\@empty
3273       \fi
3274 ⟨lua-⟩    \fi
3275   \else
3276     \MT@warn@unknown
3277 ⟨xe-⟩      \let\MT@char\@empty
3278   \fi
3279 ⟨*xe-⟩
3280   \else
```

There are more possibilities for X_ET_EX: It may be a Unicode codepoint (prefixed with 'U') or a glyph name (prefixed with '/').[4] We indicate glyph names to \MT@get@charwd by reversing the sign of \MT@char@.

```
3281     \ifnum\MT@char=47\relax
3282       \ifMT@norest \edef\MT@char{U47}%
3283       \else
3284         \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
3285         \ifnum\@tempcnta=\z@
3286           \MT@warn@unknown
3287           \let\MT@char\@empty
3288         \else
3289           \edef\MT@char{\@tempa\space}%
3290           \edef\MT@char@{-\the\@tempcnta}%
3291 ⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}%
3292         \fi
3293       \fi
3294     \else
3295       \ifnum\MT@char > \m@ne
3296         \ifMT@norest
```

Or, it's a Unicode number, which we mustn't translate into a glyph number, since the latter is font-specific. But we add the 'U' prefix.

```
3297         \@tempcnta=\XeTeXcharglyph\MT@char\relax
3298         \ifnum\@tempcnta=\z@
3299           \MT@info@missing@char
3300           \let\MT@char\@empty
3301         \else
3302 ⟨debug⟩\MT@dinfo@nl{3}{> (glyph number: \the\@tempcnta,
3303 ⟨debug⟩                   glyph name:   \XeTeXglyphname\MT@font\@tempcnta)}%
3304           \edef\MT@char{U\MT@char}%
3305         \fi
3306       \else
3307         \MT@warn@rest
3308         \let\MT@char\@empty
3309       \fi
3310     \else
3311       \MT@warn@unknown
3312       \let\MT@char\@empty
3313     \fi
```

---

4   This doesn't seem to be documented anywhere, but it has been announced here: https://tug.org/pipermail/xetex/2010-May/016531.html

```
3314     \fi
3315   \fi
3316 ⟨/xe-⟩
3317 }
3318 ⟨/pdf-|lua-|xe-⟩
```

This is the lua function to translate glyph name into slot number. Beginning with
v2.2, luaotfload provides this function in its API, which we use if available, but
(for now, at least) keep the old code for backward compatibility. With HarfBuzz, the
return value is not guaranteed to be inside the Unicode range, so we have to guard
against this case as well (same as in do_font). Also, older versions of luaotfload
(until v3.18) returned the numbers as floats.

```
3319 ⟨*luafile⟩
3320 if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
3321   local slot_of_name = luaotfload.aux.slot_of_name
3322   microtype.name_to_slot = function(name, unsafe)
3323     local n = slot_of_name(font.current(), name, unsafe)
3324     if not n then return -1 end
3325     if n > 1114111 then return -1 end
3326     return math.tointeger(n)
3327   end
3328 else
3329   -- we dig into internal structure (should be avoided)
3330   local function name_to_slot(name, unsafe)
3331     if fonts then
3332       local unicodes
3333       if fonts.ids then        -- legacy luaotfload
3334         local tfmdata = fonts.ids[font.current()]
3335         if not tfmdata then return end
3336         unicodes = tfmdata.shared.otfdata.luatex.unicodes
3337       else                      -- new location
3338         local tfmdata = fonts.hashes.identifiers[font.current()]
3339         if not tfmdata then return end
3340         unicodes = tfmdata.resources.unicodes
3341       end
3342       local unicode = unicodes[name]
3343       if unicode then -- does the 'or' branch actually exist?
3344         return type(unicode) == "number" and unicode or unicode[1]
3345       end
3346     end
3347   end
3348   microtype.name_to_slot = name_to_slot
3349 end
3350
3351 ⟨/luafile⟩
```

\MT@is@letter   Input is a letter, a character or a number.

\MT@max@char   Warning if resulting character or slot number is too large.

\MT@max@slot
```
3352 ⟨*pdf-|lua-|xe-⟩
3353 \def\MT@max@char
3354 ⟨pdf-⟩   {127 }
3355 ⟨lua-|xe-⟩   {1114111 }
3356 \def\MT@max@slot
3357 ⟨pdf-⟩   {255 }
3358 ⟨lua-|xe-⟩   {1114111 }
3359 ⟨/pdf-|lua-|xe-⟩
```

\ifMT@norest   Test whether all of the string has been used up.

```
3360 ⟨*package⟩
3361 \newif\ifMT@norest
3362 \def\MT@is@letter#1#2\relax{%
3363   \ifcat a\noexpand#1\relax
3364     \edef\MT@char@{\number`#1}%
```

```
3365    \ifx\\#2\\%
3366 ⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a letter (\MT@char@)}%
3367      \else
3368        \MT@norestfalse
3369      \fi
3370    \else
3371      \ifcat !\noexpand#1\relax
3372        \edef\MT@char@{\number`#1}%
3373 ⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a character (\MT@char@)}%
3374        \ifx\\#2\\%
3375          \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
3376        \else
3377          \MT@norestfalse
3378          \expandafter\MT@is@number#1#2\relax\relax
3379        \fi
3380      \fi
3381    \fi
3382 }
```

\MT@is@number   Numbers may be specified as a three-digit decimal number (029), as a hexadecimal
number (prefixed with ": "1D) or as a octal number (prefixed with ': '35). They
must consist of at least three characters (including the prefix), that is, "F is not
permitted.

```
3383 \def\MT@is@number#1#2#3\relax{%
3384   \ifx\relax#3\relax \else
3385     \ifx\relax#2\relax \else
3386       \MT@noresttrue
3387       \if#1"\relax
3388         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
3389 ⟨debug⟩\MT@dinfo@nl{3}{> ... a hexadecimal number: \MT@char@}%
3390       \else
3391         \if#1'\relax
3392           \def\MT@char@{\number#1#2#3}%
3393 ⟨debug⟩\MT@dinfo@nl{3}{> ... an octal number: \MT@char@}%
3394         \else
3395           \MT@ifint{#1#2#3}{%
3396             \def\MT@char@{\number#1#2#3}%
3397 ⟨debug⟩\MT@dinfo@nl{3}{> ... a decimal number: \MT@char@}%
3398           }\MT@norestfalse
3399         \fi
3400       \fi
3401       \ifnum\MT@char@ > \MT@max@slot
3402         \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
3403         \let\MT@char@\m@ne
3404       \fi
3405     \fi
3406   \fi
3407 }
```

\MT@is@active   Expand an active character. (This was completely broken in v1.7, and only worked
by chance before.) We \set@display@protect to translate, e.g., Ä into \"A, that is
to whatever it is defined in the inputenc encoding file.

   Unfortunately, the (older) inputenc definitions prefer the protected/generic
variants (e.g., \copyright instead of \textcopyright), which our parser won't
be able to understand. (I'm fed up now, so you have to complain if you really,
really want to be able to write '©' instead of \textcopyright, thus rendering your
configuration files unportable.)

   Unicode characters (inputenc/utf8,utf8x) are also supported.

```
3408 \def\MT@is@active#1#2\@nil{%
3409   \ifnum\catcode`#1 = \active
3410     \begingroup
```

```
3411        \set@display@protect
3412        \let\IeC\@firstofone
3413        \let\@inpenc@undefined@\MT@undefined@char
```

Unicode handling has changed again with LaTeX 2019/10/01.

```
3414        \let\UTF@two@octets@noexpand\@empty
3415        \let\UTF@three@octets@noexpand\@empty
3416        \let\UTF@four@octets@noexpand\@empty
```

We refrain from checking whether there is a sufficient number of octets.

```
3417        \def\UTFviii@defined##1{\ifx ##1\relax
3418          \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

For ucs (utf8x). Let's call it experimental . . .

```
3419        \MT@ifdefined@c@T\PrerenderUnicode
3420          {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
```

The \expandafter hocus-pocus should please `newunicodechar`.

```
3421        \edef\x{\endgroup
3422          \def\noexpand\@tempa{\expandafter\expandafter\expandafter\@empty\@tempa}%
```

Append what we think the translation is to the token register we use for the log.

```
3423          \MT@toks={\the\MT@toks\space(=
3424                  \expandafter\expandafter\expandafter\@empty\@tempa)}%
3425        }%
3426      \x
3427    \fi
3428 }
```

`\MT@undefined@char`   For characters not defined in the current input encoding.

```
3429 \def\MT@undefined@char#1{undefined in input encoding ``#1''}
```

`\MT@is@symbol`   The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding \⟨command⟩, we construct the command \⟨encoding⟩\⟨command⟩ and see whether its meaning is \char"⟨hex number⟩, which is the case for everything that has been defined with \DeclareTextSymbol in the encoding definition files.

```
3430 \def\MT@is@symbol{%
3431    \expandafter\def\expandafter\MT@char\expandafter
3432        {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
```

Since recently, some glyphs are defined optionally in LaTeX by checking if the glyph actually exists in the font (e.g., \textasteriskcentered).

```
3433    \expandafter\expandafter\expandafter
3434      \MT@is@opt@char\MT@char\iffontchar\char\else\fi\relax
3435    \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
3436      \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
3437    \ifnum\MT@char@ < \z@
```

In TU encoding, some commands (currently, \textquotesingle, \textasciigrave and \textquotedbl) are defined by means of the auxiliary macro \remove@tlig, which we take care of here.

```
3438        \expandafter\expandafter\expandafter\MT@is@tlig\MT@char\relax\relax
3439        \ifnum\MT@char@ < \z@
```

Finally, if it hasn't been defined by \DeclareTextSymbol, it could be a letter (e.g., \i, when using frenchpro).

```
3440          \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
3441        \fi
3442    \fi
3443 }
```

\MT@is@opt@char This seems adventurous, but we're only redefining the text command within the scope of our setup.

```
3444 \def\MT@is@opt@char#1\iffontchar#2\char#3\else#4\fi\relax{%
3445   \MT@ifempty{#1}{%
3446     \iffontchar#2%
3447       \expandafter\chardef
3448         \csname\MT@encoding\MT@detokenize@c\@tempa\endcsname=#3\relax
3449     \fi
3450   }\relax
3451 }
```

\MT@is@char A helper macro that inspects the \meaning of its argument.

\MT@charstring
```
3452 \begingroup
3453   \catcode`\/=\z@
3454   /MT@map@tlist@n{/\CHARLEX}/@makeother
3455   /lowercase{%
3456     /def/x{/endgroup
3457       /def/MT@charstring{\CHAR"}%
3458       /def/MT@is@char##1\CHAR"##2##3##4/relax{%
3459         /ifx/relax##4/relax
3460           /ifMT@xunicode
3461             /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
3462               /relax/relax/relax/relax/relax
3463           /fi
3464         /else
3465           /ifx/relax##1/relax
3466             /if##3\/relax
3467               /edef/MT@char@{/number"##2}%
3468               /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
3469             /else
3470               /edef/MT@char@{/number"##2##3}%
3471               /MT@ifstreq/MT@charstring{##4}/relax
3472                 {/MT@is@xchar##2##3|##4\CHAR"/relax}%
3473             /fi
3474 ⟨debug⟩     /MT@dinfo@nl{3}{> `/the/MT@toks' is a \char (/MT@char@)}%
3475           /fi
3476         /fi
3477       }%
```

\MT@is@xchar With fontspec's TU encoding, glyph numbers may be up to four digits.

```
3478       /def/MT@is@xchar##1|##2\CHAR"##3##4/relax{%
3479         /MT@ifstreq/MT@charstring{##3##4}%
3480           {/edef/MT@char@{/number"##1##2}}/MT@norestfalse
3481       }%
```

\MT@charxstring For xunicode, which doesn't \countdef, but rather \defs the chars.

\MT@strip@prefix
\MT@is@charx
```
3482       /def/MT@charxstring{\CHAR "}%
3483       /def/MT@strip@prefix##1>##2/relax{##2}%
3484       /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
3485         /ifx/relax##1/relax
3486           /ifx/relax##6/relax/else
3487             /edef/MT@char@{/number"##2##3##4##5}%
3488             /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@norestfalse
3489 ⟨debug⟩     /MT@dinfo@nl{3}{> `/the/MT@toks' is a xunicode \char (/MT@char@)}%
3490           /fi
3491         /fi
3492       }%
3493     }%
3494   }
3495 /x
```

\MT@is@tlig This might have to change again with the next LATEX release, ... or so I feared, but it still seems to be fine.

```
3496 \def\MT@is@tlig#1#2\relax{%
3497   \ifx\remove@tlig#1%
3498 ⟨debug⟩       \MT@dinfo@nl{3}{> `\the\MT@toks' (removing remove@tlig)}%
3499     \MT@remove@tlig
3500   \fi
3501 }
```

\MT@remove@tlig   We remove the \remove@tlig command and only pass on the number.

```
3502 \def\MT@remove@tlig{%
3503   \expandafter\MT@exp@two@c\expandafter\MT@is@number
3504   \expandafter\@secondoftwo\MT@char\relax\relax
3505 }
```

\MT@is@composite   Here, we are dealing with accented characters, specified as two tokens.

```
3506 \def\MT@is@composite#1#2\relax{%
3507   \ifx\\#2\\\else
```

Again, we construct a control sequence, this time of the form: \\⟨encoding⟩
\⟨accent⟩-⟨character⟩, e.g., \\T1\"-a, which we then expand once to see if it is a
letter (if it has been defined by \DeclareTextComposite). This should be robust,
finally, especially, since we also \detokenize the input instead of only \stringifying
it. Thus, we will die gracefully even on wrong Unicode input without utf8.

```
3508     \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
3509               \string\csname\MT@encoding\endcsname
3510               \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
```

In 2017, LATEX introduced a new way of declaring accented Unicode commands
(\DeclareUnicodeComposite), which we take care of here (\UnicodeEncodingName
has been introduced at the same time):

```
3511     \ifx\UnicodeEncodingName\@undefined\else
3512       \expandafter\expandafter\expandafter
3513         \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
3514     \fi
3515     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

Again, xunicode.

```
3516     \ifnum\MT@char@ < \z@
3517       \ifMT@xunicode
3518         \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
3519         \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
3520             \MT@char\MT@charxstring\relax\relax\relax\relax\relax
3521       \fi
3522     \fi
3523   \fi
3524 }
```

\MT@is@uni@comp   Helper for \DeclareUnicodeComposite.

```
3525 \def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
3526   \ifx\\#1\\\edef\MT@char{\iffontchar#2\fi}\fi
3527 }
```

[What about math? Well, for a moment the following looked like a solution, with
\mt@is@mathchar defined accordingly, analogous to \MT@is@char above, to pick up
the last two tokens (the \meaning of a \mathchardef'ed command expands to its
hexadecimal notation):

```
\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode`#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
```

```
    \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}
```

However, the problem is that \mathcodes and \mathchardefs have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e.g., the minus in cmsy when the euler package is loaded). So, no way to go, unfortunately.]

Some warning messages, for performance reasons separated here.

\MT@curr@list@name   The type and name of the current list, defined at various places.

\MT@set@listname
```
3528 \def\MT@set@listname{%
3529    \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
3530      `\@nameuse{MT@\MT@feat @c@name}'}%
3531 }
```

\MT@warn@ascii   For 'other' characters > 127, we issue a warning (inputenc probably hasn't been loaded), since correspondence with the slot numbers would be purely coincidental.
```
3532 \def\MT@warn@ascii{%
3533    \MT@warning@nl{Character `\the\MT@toks' (= \MT@char@)
3534      is outside of ASCII range.\MessageBreak
3535      You must load the `inputenc' package before using\MessageBreak
3536      8-bit characters in \MT@curr@list@name}%
3537 }
```

\MT@warn@number@too@large   Number too large.
```
3538 \def\MT@warn@number@too@large#1{%
3539    \MT@warning@nl{%
3540      Number #1 in encoding `\MT@encoding' too large!\MessageBreak
3541      Ignoring it in \MT@curr@list@name}%
3542 }
```

\MT@warn@rest   Not all of the string has been parsed.
```
3543 \def\MT@warn@rest{%
3544    \MT@warning@nl{%
3545      Unknown slot number of character\MessageBreak`\the\MT@toks'%
3546      \MT@warn@maybe@inputenc\MessageBreak
3547      in font encoding `\MT@encoding'.\MessageBreak
3548      Make sure it's a single character\MessageBreak
3549      (or a number) in \MT@curr@list@name}%
3550 }
```

\MT@warn@unknown   No idea what went wrong.
```
3551 \def\MT@warn@unknown{%
3552    \MT@warning@nl{%
3553      Unknown slot number of character\MessageBreak`\the\MT@toks'%
3554      \MT@warn@maybe@inputenc\MessageBreak
3555      in font encoding `\MT@encoding' in \MT@curr@list@name}%
3556 }
```

\MT@warn@maybe@inputenc   In case an input encoding had been requested.
```
3557 \def\MT@warn@maybe@inputenc{%
3558    \MT@ifdefined@n@T
3559      {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}%
3560      { (input encoding `\@nameuse
3561      {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}')}%
3562 }
```

1.2.9 Hook into LATEX's font selection

We append \MT@setupfont to \pickup@font, which is called by LATEX every time a font is selected. We then check whether we've already seen this font, and if not, set it up for micro-typography. This ensures that we will catch all fonts, and that

we will not set up fonts more than once. The whole package really hangs on this command.

In contrast to the pdfcprot package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up.

For my reference:

- \pickup@font is called by \selectfont, \wrong@fontshape, or \getanddefine@fonts (for math).

- \pickup@font calls \define@newfont.

- \define@newfont may call (inside a group!)

  – \wrong@fontshape, which in turn will call \pickup@font, and thus \define@newfont again, or

  – \extract@font.

- \get@external@font is called by \extract@font, by itself, and by the substitution macros.

Up to version 1.3 of this package, we were using \define@newfont as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before microtype and were loading fonts, e.g., jurabib, ledmac, pifont (loaded by hyperref), tipa, and probably many more. Furthermore, we had to include a hack for the IEEEtran class which loads all fonts in the class file itself (to fine tune inter-word spacing), and the memoir class, too. To cut this short: it seemed to get out of hand, and I decided that it would be better to use \pickup@font and decide for ourselves whether we've already seen that font. I hope the overhead isn't too large.

\MT@font@list We use a comma separated list.

\MT@font
```
3563 \let\MT@font@list\@empty
3564 \let\MT@font\@empty
```

All this is done at the beginning of the document. It doesn't work for plain, of course, which doesn't have \pickup@font.

```
3565 ⟨/package⟩
3566 ⟨*package | letterspace⟩
3567 ⟨plain⟩\MT@requires@latex2{
3568 \MT@addto@setup{%
```

\MT@orig@pickupfont The luatexja package redefines \char, which will upset our parsing of text symbols and commands; instead of fixing this, we won't bother, at least for the moment, but simply issue a warning and disable all further warnings. The fix is left to the user by not specifying any text commands but only (Unicode) letters. The xeCJK package, or rather its xunicode-addon, also modifies the way text symbols are defined (like luatexja but in a different way). Again, we only issue a warning.

```
3569 ⟨package⟩   \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
3570 ⟨package⟩   \MT@with@package@T{xeCJK}   {\MT@warn@unknown@once{xeCJK}}%
```

microtype also works with CJK in the sense that nothing will break when both packages are used at the same time. However, since CJK has its own way of encoding, it is currently not possible to create character-specific settings. That is, the only

feature available with CJK fonts is (non-selected) expansion. (Tracking doesn't really work for other reasons.) Like us, CJK redefines `\pickup@font`.

```
3571    \@ifpackageloaded{CJK}{%
```

The xeCJK package in turn pretends that CJK was loaded, but does not change the definition of `\pickup@font`. With xeCJK, protrusion should be possible also for C/J/K characters; I haven't tried it, though.

```
3572      \@ifpackageloaded{xeCJK}{\@firstofone}{%
3573      \@ifpackagelater{CJK}{2006/10/17}% 4.7.0
3574        {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
3575        {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
3576      \g@addto@macro\MT@orig@pickupfont
3577        {{\expandafter\ifx\font@name\relax\define@newfont\fi}}%
```

CJKutf8 redefines `\pickup@font` once more (recent versions, in PDF mode, as determined by ifpdf, which CJKutf8 loads).

```
3578      \@ifpackageloaded{CJKutf8}%
3579        {\@ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
3580          {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
3581          {\@firstoftwo}}%
3582        {\@firstoftwo}%
3583      {\g@addto@macro\MT@orig@pickupfont{%
3584        {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
3585          \define@newfont\else\xdef\font@name{%
3586            \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
3587      {\g@addto@macro\MT@orig@pickupfont{%
3588        {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
3589          \define@newfont\def\CJK@temp{v}%
3590          \ifx\CJK@temp\CJK@plane
3591            \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
3592            \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
3593          \else \CJK@addcmap\CJK@plane \fi
3594         \else\xdef\font@name{%
3595            \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
3596      \@gobble
3597    }%
3598  }{\@firstofone}%
```

This is the normal LaTeX definition.

```
3599    {\def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}}%
```

Check whether `\pickup@font` is defined as expected. The warning issued by `\CheckCommand*` would be a bit too generic.

```
3600    \ifx\pickup@font\MT@orig@pickupfont \else
3601      \MT@warning@nl{%
3602        Command \string\pickup@font\space is not defined as expected.%
3603        \MessageBreak Patching it anyway. Some things may break%
3604 ⟨*package⟩
3605        .\MessageBreak Double-check whether micro-typography is indeed%
3606        \MessageBreak applied to the document.%
3607        \MessageBreak (Hint: Turn on `verbose' mode)%
3608 ⟨/package⟩
3609      }%
3610    \fi
```

`\pickup@font`   Then we append our stuff. Everything is done inside a group.

```
3611    \g@addto@macro\pickup@font{\begingroup}%
```

If the trace package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.

```
3612    \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
3613    \g@addto@macro\pickup@font{%
3614      \escapechar\m@ne
```

```
3615 ⟨*package⟩
3616 ⟨debug⟩        \global\MT@inannottrue
3617 ⟨debug⟩        \MT@glet\MT@pdf@annot\@empty
3618 ⟨debug⟩        \MT@addto@annot{(line \number\inputlineno)}%
```

If \MT@font is empty, no substitution has taken place, hence \font@name is correct. Otherwise, if they are different, \font@name does not describe the font actually used. This test will catch first order substitutions, like bx to b, but it will still fail if the substituting font is itself substituted.

```
3619           \MT@let@cn\MT@font{MT@subst@\expandafter\string\font@name}%
3620           \ifx\MT@font\relax
3621             \let\MT@font\font@name
3622           \else
3623             \ifx\MT@font\font@name \else
3624 ⟨debug⟩     \MT@addto@annot{= substituted with \MT@@font}%
3625               \MT@register@subst@font
3626             \fi
3627           \fi
3628           \MT@setupfont
3629 ⟨/package⟩
3630 ⟨letterspace⟩        \MT@tracking
3631       \endgroup
3632    }%
3633 ⟨*package⟩
```

\MT@pickupfont
\MT@MT@pickupfont
\MT@ltx@pickupfont

Remember the patched command, because we may have to disable ourselves in certain situations.

```
3634    \let\MT@pickupfont\pickup@font
3635    \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
3636    \def\MT@ltx@pickupfont{\let\pickup@font\MT@orig@pickupfont}%
```

\do@subst@correction

Additionally, we hook into \do@subst@correction, which is called if a substitution has taken place, to record the name of the ersatz font. Unfortunately, this will only work for one-level substitutions. We have to remember the substitute for the rest of the document, not just for the first time it is called, since we need it every time a font is letterspaced.

```
3637    \g@addto@macro\do@subst@correction
3638      {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
3639        \MT@glet@nc{MT@subst@\expandafter\string\font@name}\MT@font}%
```

\add@accent
\MT@orig@add@accent

Inside \add@accent, we have to disable microtype's setup, since the grouping in the patched \pickup@font would break the accent if different fonts are used for the base character and the accent. Fortunately, LATEX takes care that the fonts used for the \accent are already set up, so that we cannot be overlooking them.

```
3640    \let\MT@orig@add@accent\add@accent
3641    \def\add@accent#1#2{%
3642      \MT@ltx@pickupfont
3643      \MT@orig@add@accent{#1}{#2}%
3644      \MT@MT@pickupfont
3645    }%
3646 ⟨/package⟩
3647 }
3648 ⟨plain⟩}\relax
3649 ⟨*package⟩
```

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\MT@check@font   Check whether we've already seen the current font.

```
3650 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

\MT@register@font — Register the current font.

```
3651 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}
```

\MT@register@subst@font — Register the substituted font (only if it isn't registered already). Additionally, we have to remove the substitute font from the list of fonts, so that we set it up again.

```
3652 \def\MT@register@subst@font{%
3653   \MT@exp@one@n\MT@in@clist\font@name\MT@font@list
3654   \ifMT@inlist@\else
3655     \xdef\MT@font@list{\MT@font@list\font@name,}%
3656     \expandafter\MT@rem@from@clist\MT@font\MT@font@list
3657   \fi
3658 }
```

1.2.10 Context-sensitive setup

Here are the variants for context-sensitive setup.

\MT@active@features — The activated features are stored in this command.

```
3659 \let\MT@active@features\@empty
```

\MT@check@font@cx — Every feature has its own list of fonts that have already been dealt with. If the font needn't be set up for a feature, we temporarily disable the corresponding setup command. This should be more efficient than book-keeping the fonts in lists associated with the combination of contexts, as we've done it before.

```
3660 \def\MT@check@font@cx{%
3661   \MT@if@true
3662   \MT@map@clist@c\MT@active@features{%
3663     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
3664       \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
3665     \ifMT@inlist@
3666       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
3667     \else
3668       \MT@if@false
3669     \fi
3670   }%
3671   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
3672 }
```

\MT@register@subst@font@cx — Add the substituted font to each feature list and possibly remove substitute font.

```
3673 \def\MT@register@subst@font@cx{%
3674   \MT@map@clist@c\MT@active@features{%
3675     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
3676       \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
3677     \ifMT@inlist@ \else
3678       \MT@exp@cs\MT@xadd
3679         {MT@##1@\csname MT@##1@context\endcsname font@list}%
3680         {\font@name,}%
3681       \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font
3682         \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
3683     \fi
3684   }%
3685 }
```

\MT@register@font@cx — For each feature, add the current font to the list, unless we didn't set it up.

```
3686 \def\MT@register@font@cx{%
3687   \MT@map@clist@c\MT@active@features{%
3688     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
3689       \MT@exp@cs\MT@xadd
3690         {MT@##1@\csname MT@##1@context\endcsname font@list}%
3691         {\MT@font,}%
3692       \def\@tempa{##1}%
3693       \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
```

```
3694      \fi
3695    }%
3696 }
```

\MT@maybe@rem@from@list   Recurse through all context font lists of the document and remove the font, unless
it's the current context.

```
3697 \def\MT@maybe@rem@from@list#1{%
3698   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
3699     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
3700       \MT@font \csname MT@\@tempa @#1font@list\endcsname
3701   }%
3702 }
```

\microtypecontext   The user may change the context, so that different setups are possible. This is
\MT@microtypecontext   especially useful for multi-lingual documents.
                                  Inside the preamble, this command shouldn't actually do anything but remem-
ber itself for later.

```
3703 \def\microtypecontext{\MT@begin@catcodes\MT@microtypecontext}
3704 \def\MT@microtypecontext#1{\MT@end@catcodes\MT@addto@setup{\microtypecontext{#1}}}
3705 \MT@addto@setup{%
3706   \DeclareRobustCommand\microtypecontext{%
3707     \MT@begin@catcodes
3708     \MT@microtypecontext
3709   }%
3710   \def\MT@microtypecontext#1{%
3711     \MT@end@catcodes
3712     \MT@setup@contexts
3713     \let\MT@reset@context\relax
```

We need to ensure that math fonts are set up anew.

```
3714     \MT@glet\glb@currsize\@empty
3715     \setkeys{MTC}{#1}%
3716     \selectfont
3717     \MT@reset@context
3718   }%
3719 }
```

\textmicrotypecontext   This is just a wrapper around \microtypecontext.
\MT@textmicrotypecontext
```
3720 \DeclareRobustCommand\textmicrotypecontext{\MT@begin@catcodes\MT@textmicrotypecontext}
```
\MT@text@microtypecontext
```
3721 \def\MT@textmicrotypecontext#1{\MT@end@catcodes\MT@text@microtypecontext{#1}}
3722 \def\MT@text@microtypecontext#1#2{{\microtypecontext{#1}#2}}
```

\MT@reset@context   We have to reset the font at the end of the group, provided there actually was a
\MT@reset@context@   change.

```
3723 \def\MT@reset@context@{%
3724   \MT@vinfo{<<< Resetting contexts\on@line
3725 ⟨debug⟩   \MessageBreak= \MT@pr@context/\MT@ex@context
3726 ⟨debug⟩                     /\MT@tr@context/\MT@kn@context/\MT@sp@context
3727   }%
3728   \selectfont
3729 }
```

\MT@setup@contexts   The first time \microtypecontext is called, we initialise the context lists and
redefine the commands used in \pickup@font.

```
3730 \def\MT@setup@contexts{%
3731   \MT@map@clist@c\MT@active@features
3732     {\MT@glet@nc{MT@##1@@font@list}\MT@font@list}%
3733   \MT@glet\MT@check@font\MT@check@font@cx
3734   \MT@glet\MT@register@font\MT@register@font@cx
3735   \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
3736   \MT@glet\MT@setup@contexts\relax
3737 }
```

Define context keys.

```
3738 \MT@map@clist@c\MT@features@long{%
3739   \define@key{MTC}{#1}[]{%
3740     \edef\@tempb{\@nameuse{MT@rbba@#1}}%
3741     \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
3742     \ifMT@inlist@
```

Using an empty context is only asking for trouble, therefore we choose the '@' instead (hoping for the LaTeX users' natural awe of this character).

```
3743       \MT@ifempty{##1}{\def\MT@val{@}}{\def\MT@val{##1}}%
3744       \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
3745 ⟨debug⟩\MT@dinfo{1}{>>> no change of #1 context: `\MT@val'}%
3746       \else
3747         \MT@vinfo{>>> Changing #1 context to `\MT@val'\MessageBreak\on@line
3748 ⟨debug⟩              \space(previous: `\@nameuse{MT@\@tempb @context}')%
3749                    }%
3750         \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

The next time we see the font, we have to reset *all* factors.

```
3751         \MT@glet@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%
```

We must also keep track of all contexts in the document.

```
3752         \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
3753           \MT@val \csname MT@\@tempb @doc@contexts\endcsname
3754         \ifMT@inlist@ \else
3755           \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{{\MT@val}}%
3756 ⟨debug⟩    \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
3757         \fi
3758         \MT@edef@n{MT@\@tempb @context}{\MT@val}%
3759       \fi
3760     \fi
3761   }%
3762 }
```

We also allow the `activate` shortcut.

```
3763 \define@key{MTC}{activate}[]{%
3764   \setkeys{MTC}{protrusion={#1}}%
3765   \setkeys{MTC}{expansion={#1}}%
3766 }
```

<div style="float:left">

\MT@pr@context
\MT@ex@context
\MT@tr@context
\MT@sp@context
\MT@kn@context
\MT@pr@doc@contexts
\MT@ex@doc@contexts
\MT@tr@doc@contexts
\MT@sp@doc@contexts
\MT@kn@doc@contexts
\DeclareMicrotypeSet
\MT@extra@context
\DeclareMicrotypeSet*

</div>

Initialise the contexts.

```
3767 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
3768   \MT@def@n{MT@#1@context}{@}%
3769   \MT@def@n{MT@#1@doc@contexts}{{@}}%
3770 }
3771 \let\MT@extra@context\@empty
```

1.3 Configuration

1.3.1 Font sets

Calling this macro will create a comma list for every font attribute of the form: \MT⟨*feature*⟩list@⟨*attribute*⟩@⟨*set name*⟩. If the optional argument is empty, lists for all available features will be created.

The third argument must be a list of `key=value` pairs. If a font attribute is not specified, we define the corresponding list to \relax, so that it does not constitute a constraint.

```
3772 \def\DeclareMicrotypeSet{%
3773   \MT@begin@catcodes
3774   \@ifstar
3775     \MT@DeclareSetAndUseIt
```

```
3776        \MT@DeclareSet
3777 }
```

\MT@DeclareSet

```
3778 \newcommand\MT@DeclareSet[3][]{%
3779   \MT@ifempty{#1}{%
3780     \MT@map@clist@c\MT@features{\begingroup\MT@declare@sets{##1}{#2}{#3}\endgroup}%
3781   }{%
3782     \MT@map@clist@n{#1}{\begingroup
3783       \MT@ifempty{##1}\relax{%
3784         \MT@is@feature{##1}{set declaration `#2'}{%
3785           \MT@exp@one@n\MT@declare@sets
3786             {\csname MT@rbba@##1\endcsname}{#2}{#3}%
3787         }%
3788       }%
3789     \endgroup}%
3790   }%
3791   \MT@end@catcodes
3792 }
```

\MT@DeclareSetAndUseIt

```
3793 \newcommand\MT@DeclareSetAndUseIt[3][]{%
3794   \MT@DeclareSet[#1]{#2}{#3}%
3795   \UseMicrotypeSet[#1]{#2}%
3796 }
```

\MT@curr@set@name   We need to remember the name of the set currently being declared.

```
3797 \let\MT@curr@set@name\@empty
```

\MT@declare@sets   Define the current set name and parse the keys.

```
3798 \def\MT@declare@sets#1#2#3{%
3799   \def\MT@curr@set@name{#2}%
3800   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
3801     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
3802     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
3803       \MT@glet@nc{MT@#1list@##1@\MT@curr@set@name}\@undefined
3804     }%
3805   }%
3806   \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
3807 ⟨debug⟩\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
3808   \setkeys{MT@#1@set}{#3}%
3809 }
```

\MT@define@set@key@   ⟨#1⟩ = font axis, ⟨#2⟩ = feature.

```
3810 \def\MT@define@set@key@#1#2{%
3811   \define@key{MT@#2@set}{#1}[]{%
3812     \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
3813     \MT@map@clist@n{##1}{%
3814       \KV@@sp@def\MT@val{####1}%
3815       \MT@get@highlevel{#1}%
```

We do not add the expanded value to the list . . .

```
3816       \MT@exp@two@n\g@addto@macro
3817         {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
3818         {\MT@val,}%
3819     }%
```

. . . but keep in mind that the list has to be expanded at the end of the preamble.

```
3820     \expandafter\g@addto@macro\expandafter\MT@font@sets
3821       \csname MT@#2list@#1@\MT@curr@set@name\endcsname
3822 ⟨debug⟩\MT@dinfo@nl{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
3823   }%
3824 }
```

\MT@get@highlevel   Saying, for instance, 'family=rm*' or 'shape=bf*' will expand to \rmdefault resp.
                     \bfdefault.

```
3825 \def\MT@get@highlevel#1{%
3826   \expandafter\MT@test@ast\MT@val*\@nil\relax{%
```

And 'family = *' will become \familydefault.

```
3827     \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
```

Test whether the command is actually defined.

```
3828     \MT@ifdefined@n@TF{\@tempa default}%
3829       {\edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}}%
3830       {\MT@warning{`\@backslashchar\@tempa default' is not a defined command.\MessageBreak
3831                    Ignoring `#1 = {\@tempa*}' in font set\MessageBreak`\MT@curr@set@name'}%
3832        \let\MT@val\@empty}%
```

In contrast to earlier versions, these values will not be expanded immediately, but
at the end of the preamble.

```
3833   }%
3834 }
```

\MT@test@ast   It the last character is an asterisk, execute the second argument, otherwise the first
                    one.

```
3835 \def\MT@test@ast#1*#2\@nil{%
3836   \def\@tempa{#1}%
3837   \MT@ifempty{#2}%
3838 }
```

\MT@font@sets   Fully expand the font specification and fix catcodes for all font sets. Also remove
\MT@fix@font@set   fontspec's counters.

```
3839 \let\MT@font@sets\@empty
3840 \def\MT@fix@font@set#1{%
3841   \MT@ifdefined@c@T{#1}{%
3842     \xdef#1{#1}%
3843     \ifMT@fontspec
3844       \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
3845     \fi
3846     \global\@onelevel@sanitize#1%
3847   }%
3848 }
```

\MT@define@set@key@size   size requires special treatment.

```
3849 \def\MT@define@set@key@size#1{%
3850   \define@key{MT@#1@set}{size}[]{%
3851     \MT@map@clist@n{##1}{%
3852       \def\MT@val{####1}%
3853       \expandafter\MT@get@range\MT@val--\@nil
3854       \ifx\MT@val\relax \else
3855         \MT@exp@cs\MT@xadd
3856           {MT@#1list@size@\MT@curr@set@name}%
3857           {{{\MT@lower}{\MT@upper}\relax}}%
3858       \fi
3859     }%
3860 ⟨debug⟩\MT@dinfo@nl{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
3861   }%
3862 }
```

Font sizes may also be specified as ranges. This has been requested by Andreas
Bühmann, who has also offered valuable help in implementing this. Now, it is for
instance possible to set up different lists for fonts with optical sizes. (The MinionPro
project does this for the OpenType version of Adobe's Minion. (Available from CTAN
at pkg/minionpro))

`\MT@get@range`
`\MT@upper`
`\MT@lower`

Ranges will be stored as triplets of {⟨*lower bound*⟩}{⟨*upper bound*⟩}{⟨*list name*⟩}.
For simple sizes, the upper boundary is −1.

```
3863 \def\MT@get@range#1-#2-#3\@nil{%
3864   \MT@ifempty{#1}{%
3865     \MT@ifempty{#2}{%
3866       \let\MT@val\relax
3867     }{%
3868       \def\MT@lower{0}%
3869       \def\MT@val{#2}%
3870       \MT@get@size
3871       \edef\MT@upper{\MT@val}%
3872     }%
3873   }{%
3874     \def\MT@val{#1}%
3875     \MT@get@size
3876     \ifx\MT@val\relax \else
3877       \edef\MT@lower{\MT@val}%
3878       \MT@ifempty{#2}{%
3879         \MT@ifempty{#3}%
3880           {\def\MT@upper{-1}}%
```

2048 pt is TEX's maximum font size.

```
3881           {\def\MT@upper{2048}}%
3882       }{%
3883         \def\MT@val{#2}%
3884         \MT@get@size
3885         \ifx\MT@val\relax \else
3886           \MT@ifdim\MT@lower>\MT@val{%
3887             \MT@error{%
3888               Invalid size range (\MT@lower\space > \MT@val) in font set
3889               `\MT@curr@set@name'.\MessageBreak Swapping sizes}{}%
3890             \edef\MT@upper{\MT@lower}%
3891             \edef\MT@lower{\MT@val}%
3892           }{%
3893             \edef\MT@upper{\MT@val}%
3894           }%
3895           \MT@ifdim\MT@lower=\MT@upper
3896             {\def\MT@upper{-1}}%
3897             \relax
3898         \fi
3899       }%
3900     \fi
3901   }%
3902 }
```

`\MT@get@size`   Translate a size selection command and normalise it.

```
3903 \def\MT@get@size{%
```

A single star would mean `\sizedefault`, which doesn't exist, so we define it to be
`\normalsize`.

```
3904   \if*\MT@val\relax
3905     \def\@tempa{\normalsize}%
3906   \else
3907     \MT@let@cn\@tempa{\MT@val}%
3908   \fi
3909   \ifx\@tempa\relax\else
3910     \MT@get@size@
3911   \fi
```

Test whether we finally got a number or dimension so that we can strip the 'pt'
(`\@defaultunits` and `\strip@pt` are kernel macros).

```
3912   \MT@ifdimen\MT@val{%
3913     \@defaultunits\@tempdima\MT@val pt\relax\@nnil
3914     \edef\MT@val{\strip@pt\@tempdima}%
```

```
3915    }{%
3916      \MT@warning{Could not parse font size `\MT@val'\MessageBreak
3917               in font set `\MT@curr@set@name'}%
3918      \let\MT@val\relax
3919    }%
3920 }
```

\MT@get@size@ The `relsize` solution of parsing `\@setfontsize` does not work with the AMS
\MT@get@size@@ classes, among others. I hope my hijacking doesn't do any harm. We redefine
`\set@fontsize` instead of `\@setfontsize` because some classes might define the
size selection commands by simply using `\fontsize` (e.g., the a0poster class).

```
3921 \def\MT@get@size@@{%
3922    \begingroup
3923      \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
3924      \@tempa\@nil
3925 }
```

The `svjour3` class defines the size commands using conditionals; using e-TeX
primitives, we close any leftovers here.

```
3926 ^^X\@ifclassloaded{svjour3}{%
3927 ^^X  \def\MT@get@size@{%
3928 ^^X    \@tempcnta=\currentiflevel
3929 ^^X    \MT@get@size@@
3930 ^^X    \MT@loop
3931 ^^X      \ifnum\numexpr\currentiflevel-1>\@tempcnta
3932 ^^X      \csname fi\endcsname
3933 ^^X    \MT@repeat
3934 ^^X  }%
3935 ^^X}{%
3936    \let\MT@get@size@\MT@get@size@@
3937 ^^X}
```

\MT@define@set@key@font

```
3938 \def\MT@define@set@key@font#1{%
3939    \define@key{MT@#1set}{font}[]{%
3940      \MT@glet@nc{MT@#1list@font@\MT@curr@set@name}\@empty
3941      \MT@map@clist@n{##1}{%
3942        \def\MT@val{####1}%
3943        \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
3944        \expandafter\MT@get@font\MT@val/////\@nil
3945        \MT@exp@two@n\g@addto@macro
3946          {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
3947          {\MT@val,}%
3948      }%
3949      \expandafter\g@addto@macro\expandafter\MT@font@sets
3950        \csname MT@#1list@font@\MT@curr@set@name\endcsname
3951 ⟨debug⟩\MT@dinfo@nl{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
3952    }%
3953 }
```

\MT@get@font Translate any asterisks.

```
3954 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
3955    \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
3956    \ifx\MT@val\relax\def\MT@val{0}\fi
3957    \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
3958    \let\MT@val\@tempb
3959 }
```

\MT@get@font@ Helper macro, also used by `\MT@get@font@and@size`.

```
3960 \def\MT@get@font@#1#2#3#4#5#6{%
3961    \let\@tempb\@empty
3962    \def\MT@temp{#1/#2/#3/#4/#5}%
3963    \MT@get@axis{encoding}{#1}%
```

```
3964    \MT@get@axis{family}   {#2}%
3965    \MT@get@axis{series}   {#3}%
3966    \MT@get@axis{shape}    {#4}%
3967    \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
3968    \MT@ifempty{#5}{%
3969      \MT@warn@axis@empty{size}{\string\normalsize}%
3970      \def\MT@val{*}%
3971    }{%
3972      \def\MT@val{#5}%
3973    }%
3974    \MT@get@size
3975 }
```

\MT@get@axis

```
3976 \def\MT@get@axis#1#2{%
3977    \def\MT@val{#2}%
3978    \MT@get@highlevel{#1}%
3979    \MT@ifempty\MT@val{%
3980      \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
3981      \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
3982    }\relax
3983    \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
3984 }
```

\MT@warn@axis@empty

```
3985 \def\MT@warn@axis@empty#1#2{%
3986    \MT@warning{#1 axis is empty in font specification\MessageBreak
3987      `\MT@temp'. Using `#2' instead}%
3988 }
```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

```
3989 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
3990    \MT@define@set@key@{encoding}{#1}%
3991    \MT@define@set@key@{family}  {#1}%
3992    \MT@define@set@key@{series}  {#1}%
3993    \MT@define@set@key@{shape}   {#1}%
3994    \MT@define@set@key@size      {#1}%
3995    \MT@define@set@key@font      {#1}%
3996 }
```

\UseMicrotypeSet   To use a particular set we simply redefine MT@⟨feature⟩@setname. If the optional argument is empty, set names for all features will be redefined.

```
3997 \def\UseMicrotypeSet{%
3998    \MT@begin@catcodes
3999    \MT@UseMicrotypeSet
4000 }
```

\MT@UseMicrotypeSet

```
4001 \newcommand*\MT@UseMicrotypeSet[2][]{%
4002    \MT@ifempty{#1}{%
4003      \MT@map@clist@c\MT@features{\begingroup\MT@use@set{##1}{#2}\endgroup}%
4004    }{%
4005      \MT@map@clist@n{#1}{\begingroup
4006        \MT@ifempty{##1}\relax{%
4007          \MT@is@feature{##1}{activation of set `#2'}{%
4008            \MT@exp@one@n\MT@use@set
4009              {\csname MT@rbba@##1\endcsname}{#2}%
4010          }%
4011        }%
4012      \endgroup}%
4013    }%
4014    \MT@end@catcodes
4015 }
```

\MT@pr@setname   Only use sets that have been declared.

\MT@ex@setname 4016 \def\MT@use@set#1#2{%
\MT@tr@setname 4017   \MT@ifdefined@n@TF{MT@#1set@@#2}{%
                4018   \MT@xdef@n{MT@#1setname}{#2}%
\MT@sp@setname 4019   }{%
\MT@kn@setname 4020   \MT@ifdefined@n@TF{MT@#1setname}\relax{%
    \MT@use@set 4021   \MT@xdef@n{MT@#1setname}{\@nameuse{MT@default@#1set}}%
                4022   }%
                4023   \MT@error{%
                4024   The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
                4025   Using set `\@nameuse{MT@#1setname}' instead}{}%
                4026   }%
                4027 }

\DeclareMicrotypeSetDefault   This command can be used in the main configuration file to declare the default
                                     font set, in case no set is specified in the package options.

                4028 \def\DeclareMicrotypeSetDefault{%
                4029   \MT@begin@catcodes
                4030   \MT@DeclareMicrotypeSetDefault
                4031 }

\MT@DeclareMicrotypeSetDefault

                4032 \newcommand*\MT@DeclareMicrotypeSetDefault[2][]{%
                4033   \MT@ifempty{#1}{%
                4034   \MT@map@clist@c\MT@features{\begingroup\MT@set@default@set{##1}{#2}\endgroup}%
                4035   }{%
                4036   \MT@map@clist@n{#1}{\begingroup
                4037   \MT@ifempty{##1}\relax{%
                4038   \MT@is@feature{##1}{declaration of default set `#2'}{%
                4039   \MT@exp@one@n\MT@set@default@set
                4040   {\csname MT@rbba@##1\endcsname}{#2}%
                4041   }%
                4042   }%
                4043   \endgroup}%
                4044   }%
                4045   \MT@end@catcodes
                4046 }

\MT@default@pr@set
\MT@default@ex@set 4047 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 4048   \MT@ifdefined@n@TF{MT@#1set@@#2}{%
                4049 ⟨debug⟩\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
\MT@default@sp@set 4050   \MT@xdef@n{MT@default@#1set}{#2}%
\MT@default@kn@set 4051   }{%
\MT@set@default@set 4052   \MT@error{%
                4053   The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
                4054   Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
                4055   \MT@xdef@n{MT@default@#1set}{all}%
                4056   }%
                4057 }

1.3.2 Variants and aliases

\DeclareMicrotypeVariants   Specify suffixes for variants (see fontname/variants.map). The starred version
\MT@variants   appends to the list.

                4058 \let\MT@variants\@empty
                4059 \def\DeclareMicrotypeVariants{%
                4060   \MT@begin@catcodes
                4061   \@ifstar
                4062   \MT@DeclareVariants
                4063   {\let\MT@variants\@empty\MT@DeclareVariants}%
                4064 }

\MT@DeclareVariants

```
4065 \def\MT@DeclareVariants#1{%
4066   \MT@map@clist@n{#1}{%
4067     \def\@tempa{##1}%
4068     \@onelevel@sanitize\@tempa
4069     \xdef\MT@variants{\MT@variants{\@tempa}}%
4070   }%
4071   \MT@end@catcodes
4072 }
```

\DeclareMicrotypeAlias   This can be used to set an alias name for a font, so that the file and the settings for the aliased font will be loaded.

```
4073 \def\DeclareMicrotypeAlias{%
4074   \MT@begin@catcodes
4075   \MT@DeclareMicrotypeAlias
4076 }
```

\MT@DeclareMicrotypeAlias

```
4077 \newcommand*\MT@DeclareMicrotypeAlias[2]{%
4078   \def\@tempb{#2}%
4079   \@onelevel@sanitize\@tempb
4080   \MT@ifdefined@n@T{MT@#1@alias}{%
4081     \MT@warning{Alias font family `\@tempb' will override
4082       alias `\@nameuse{MT@#1@alias}'\MessageBreak
4083       for font family `#1'}}%
4084   \MT@xdef@n{MT@#1@alias}{\@tempb}%
```

If we encounter this command while a font is being set up, we also set the alias for the current font so that if \DeclareMicrotypeAlias has been issued inside a configuration file, the configuration file for the alias font will be loaded, too.

```
4085   \MT@ifdefined@c@T\MT@family{%
4086 ⟨debug⟩\MT@dinfo{1}{Activating alias font `\@tempb' for `\MT@family'}%
4087     \MT@glet\MT@familyalias\@tempb
4088   }%
4089   \MT@end@catcodes
4090 }
```

\LoadMicrotypeFile   May be used to load a configuration file manually.

```
4091 \def\LoadMicrotypeFile#1{%
4092   \edef\@tempa{\zap@space#1 \@empty}%
4093   \@onelevel@sanitize\@tempa
4094   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
4095   \ifMT@inlist@
4096     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
4097   \else
4098     \MT@xadd\MT@file@list{\@tempa,}%
4099     \MT@begin@catcodes
4100     \InputIfFileExists{mt-\@tempa.cfg}{%
4101       \edef\MT@curr@file{mt-\@tempa.cfg}%
4102       \MT@vinfo{... Loading configuration file \MT@curr@file}%
4103     }{%
4104       \MT@warning{Configuration file mt-\@tempa.cfg\MessageBreak
4105                   does not exist}%
4106     }%
4107     \MT@end@catcodes
4108   \fi
4109 }
4110 ⟨/package⟩
4111 ⟨/package|letterspace⟩
```

1.3.3 Disabling ligatures

`\DisableLigatures`
`\MT@DisableLigatures`
`\MT@nl@setname`
`\MT@nl@ligatures`

This is really simple now: we can re-use the set definitions of `\DeclareMicrotypeSet`; there can only be one set, which we'll call 'no ligatures'.

The optional argument may be used to disable selected ligatures only.

```
4112 ⟨*pdf-|lua-⟩
4113 ⟨pdf-⟩\MT@requires@pdftex5{
4114 \def\DisableLigatures{%
4115   \MT@begin@catcodes
4116   \MT@DisableLigatures
4117 }
4118 \newcommand*\MT@DisableLigatures[2][]{%
4119   \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
4120   \xdef\MT@active@features{\MT@active@features,nl}%
4121   \global\MT@noligaturestrue
4122   \MT@declare@sets{nl}{no ligatures}{#2}%
4123   \gdef\MT@nl@setname{no ligatures}%
4124   \MT@end@catcodes
4125 }
4126 ⟨pdf-⟩}{
4127 ⟨/pdf-|lua-⟩
```

If pdfTeX is too old, we throw an error.

```
4128 ⟨*pdf-|xe-⟩
4129 \renewcommand*\DisableLigatures[2][]{%
4130   \MT@error{Disabling ligatures of a font is only possible\MessageBreak
4131     with pdftex version 1.30 or newer.\MessageBreak
4132     Ignoring \string\DisableLigatures}{%
4133 ⟨pdf-⟩    Upgrade
4134 ⟨xe-⟩    Use
4135     pdftex.}%
4136 }
4137 ⟨pdf-⟩}
4138 ⟨/pdf-|xe-⟩
```

1.3.4 Interaction with `babel`

`\DeclareMicrotypeBabelHook`

Declare the context that should be loaded when a `babel` language is selected. The command will not check whether a previous declaration will be overwritten.

```
4139 ⟨*package⟩
4140 \def\DeclareMicrotypeBabelHook#1#2{%
4141   \MT@map@clist@n{#1}{%
4142     \KV@@sp@def\@tempa{##1}%
4143     \MT@gdef@n{MT@babel@\@tempa}{#2}%
4144   }%
4145 }
```

1.3.5 Fine tuning

The commands `\SetExpansion` and `\SetProtrusion` provide an interface for setting the character protrusion resp. expansion factors for a set of fonts.

`\SetProtrusion`

This macro accepts three arguments: [options,] set of font attributes and list of character protrusion factors.

A new macro called `\MT@pr@c@`⟨*name*⟩ will be defined to be ⟨*#3*⟩ (i.e., the list of characters, not expanded).

```
4146 \def\SetProtrusion{%
4147   \MT@begin@catcodes
4148   \MT@SetProtrusion
4149 }
```

\MT@SetProtrusion   We want the catcodes to be correct even if this is called in the preamble.

\MT@pr@c@name   4150 `\newcommand*\MT@SetProtrusion[3][]{%`

\MT@extra@context   4151 `    \let\MT@extra@context\@empty`

\MT@permutelist   Parse the optional first argument. We first have to know the name before we can deal with the extra options.

4152 `    \MT@set@named@keys{MT@pr@c}{#1}%`
4153 ⟨*debug*⟩`\MT@dinfo{1}{creating protrusion list `\MT@pr@c@name'}%`
4154 `    \def\MT@permutelist{pr@c}%`
4155 `    \setkeys{MT@cfg}{#2}%`

We have parsed the second argument, and can now define macros for all permutations of the font attributes to point to \MT@pr@c@⟨*name*⟩, ...

4156 `    \MT@permute`

... which we can now define to be ⟨*#3*⟩. Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

4157 `    \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%`
4158 `    \MT@end@catcodes`
4159 `}`
4160 ⟨*/package*⟩

\SetExpansion   \SetExpansion only differs in that it allows some extra options (stretch, shrink, step, auto).

4161 ⟨**pdf-|lua-*⟩
4162 `\def\SetExpansion{%`
4163 `    \MT@begin@catcodes`
4164 `    \MT@SetExpansion`
4165 `}`

\MT@SetExpansion

\MT@ex@c@name   4166 `\newcommand*\MT@SetExpansion[3][]{%`

\MT@extra@context   4167 `    \let\MT@extra@context\@empty`

\MT@permutelist   4168 `    \MT@set@named@keys{MT@ex@c}{#1}%`
4169 `    \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @factor}{%`
4170 `      \ifnum\csname MT@ex@c@\MT@ex@c@name @factor\endcsname > \@m`
4171 `        \MT@warning@nl{Expansion factor \number\@nameuse{MT@ex@c@\MT@ex@c@name @factor}`
4172 `          too large in list\MessageBreak `\MT@ex@c@name'. Setting it to the`
4173 `          maximum of 1000}%`
4174 `        \MT@glet@nc{MT@ex@c@\MT@ex@c@name @factor}\@m`
4175 `      \fi`
4176 `    }%`
4177 ⟨*debug*⟩`\MT@dinfo{1}{creating expansion list `\MT@ex@c@name'}%`
4178 `    \def\MT@permutelist{ex@c}%`
4179 `    \setkeys{MT@cfg}{#2}%`
4180 `    \MT@permute`
4181 `    \MT@gdef@n{MT@ex@c@\MT@ex@c@name}{#3}%`
4182 `    \MT@end@catcodes`
4183 `}`

\SetTracking

4184 `\def\SetTracking{%`
4185 `    \MT@begin@catcodes`
4186 `    \MT@SetTracking`
4187 `}`

\MT@SetTracking   Third argument may be empty.

4188 `\newcommand*\MT@SetTracking[3][]{%`
4189 `    \let\MT@extra@context\@empty`
4190 `    \MT@set@named@keys{MT@tr@c}{#1}%`
4191 ⟨*debug*⟩`\MT@dinfo{1}{creating tracking list `\MT@tr@c@name'}%`
4192 `    \def\MT@permutelist{tr@c}%`
4193 `    \setkeys{MT@cfg}{#2}%`

```
4194    \MT@permute
4195    \KV@@sp@def\@tempa{#3}%
4196    \MT@ifempty\@tempa\relax{%
4197      \MT@ifint\@tempa
4198        {\MT@xdef@n{MT@tr@c@\MT@tr@c@name}{\@tempa}}%
4199        {\MT@warning{Value `\@tempa' is not a number in\MessageBreak
4200                     tracking set `\MT@curr@set@name'}}}%
4201    \MT@end@catcodes
4202  }
4203  ⟨/pdf-|lua-⟩
```

\SetExtraSpacing

```
4204  ⟨*pdf-⟩
4205  \def\SetExtraSpacing{%
4206    \MT@begin@catcodes
4207    \MT@SetExtraSpacing
4208  }
```

\MT@SetExtraSpacing
\MT@sp@c@name
\MT@extra@context
\MT@permutelist

```
4209  \newcommand*\MT@SetExtraSpacing[3][]{%
4210    \let\MT@extra@context\@empty
4211    \MT@set@named@keys{MT@sp@c}{#1}%
4212  ⟨debug⟩\MT@dinfo{1}{creating spacing list `\MT@sp@c@name'}%
4213    \def\MT@permutelist{sp@c}%
4214    \setkeys{MT@cfg}{#2}%
4215    \MT@permute
4216    \MT@gdef@n{MT@sp@c@\MT@sp@c@name}{#3}%
4217    \MT@end@catcodes
4218  }
```

\SetExtraKerning

```
4219  \def\SetExtraKerning{%
4220    \MT@begin@catcodes
4221    \MT@SetExtraKerning
4222  }
```

\MT@SetExtraKerning
\MT@kn@c@name
\MT@extra@context
\MT@permutelist

```
4223  \newcommand*\MT@SetExtraKerning[3][]{%
4224    \let\MT@extra@context\@empty
4225    \MT@set@named@keys{MT@kn@c}{#1}%
4226  ⟨debug⟩\MT@dinfo{1}{creating kerning list `\MT@kn@c@name'}%
4227    \def\MT@permutelist{kn@c}%
4228    \setkeys{MT@cfg}{#2}%
4229    \MT@permute
4230    \MT@gdef@n{MT@kn@c@\MT@kn@c@name}{#3}%
4231    \MT@end@catcodes
4232  }
4233  ⟨/pdf-⟩
```

\MT@set@named@keys
\MT@options

We first set the name (if specified), then remove it from the list, and set the remaining keys.

```
4234  ⟨*package⟩
4235  \def\MT@set@named@keys#1#2{%
4236    \def\x##1name=##2,##3\@nil{%
4237      \setkeys{#1}{name=##2}%
4238      \gdef\MT@options{##1##3}%
4239      \MT@rem@from@clist{name=}\MT@options
4240    }%
4241    \x#2,name=,\@nil
4242    \@expandtwoargs\setkeys{#1}\MT@options
4243  }
```

\MT@define@code@key

Define the keys for the configuration lists (which are setting the codes, in pdfTeX speak).

```
4244 \def\MT@define@code@key#1#2{%
4245   \define@key{MT@#2}{#1}[]{%
4246     \@tempcnta=\@ne
4247     \MT@map@clist@n{##1}{%
4248       \KV@@sp@def\MT@val{####1}%
```

Here, too, we allow for something like 'bf*'. It will be expanded immediately.

```
4249       \MT@get@highlevel{#1}%
4250       \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
4251       \advance\@tempcnta \@ne
4252     }%
4253   }%
4254 }
```

`\MT@define@code@key@family`   Remove fontspec's internal feature counter.

```
4255 \def\MT@define@code@key@family#1{%
4256   \define@key{MT@#1}{family}[]{%
4257     \@tempcnta=\@ne
4258     \MT@map@clist@n{##1}{%
4259       \KV@@sp@def\MT@val{####1}%
4260       \MT@get@highlevel{family}%
4261       \ifMT@fontspec
4262         \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
4263       \fi
4264       \MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}%
4265       \advance\@tempcnta \@ne
4266     }%
4267   }%
4268 }
```

`\MT@define@code@key@size`   \MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.

```
4269 \def\MT@define@code@key@size#1{%
4270   \define@key{MT@#1}{size}[]{%
4271     \MT@map@clist@n{##1}{%
4272       \KV@@sp@def\MT@val{####1}%
4273       \expandafter\MT@get@range\MT@val--\@nil
4274       \ifx\MT@val\relax \else
4275         \MT@exp@cs\MT@xadd{MT@tempsize}%
4276           {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
4277       \fi
4278     }%
4279   }%
4280 }
```

`\MT@define@code@key@font`

```
4281 \def\MT@define@code@key@font#1{%
4282   \define@key{MT@#1}{font}[]{%
4283     \MT@map@clist@n{##1}{%
4284       \KV@@sp@def\MT@val{####1}%
4285       \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
4286       \expandafter\MT@get@font@and@size\MT@val/////\@nil
4287       \ifMT@fontspec
4288         \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
4289       \fi
4290       \MT@xdef@n{MT@\MT@permutelist @\@tempb\MT@extra@context}%
4291         {\csname MT@\MT@permutelist @name\endcsname}%
4292 ⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempb=\MT@val
4293 ⟨debug⟩                 \ifx\MT@extra@context\@empty\else\MessageBreak
4294 ⟨debug⟩                   (context: \MT@extra@context)\fi}%
4295       \MT@exp@cs\MT@xaddb
4296         {MT@\MT@permutelist @\@tempb\MT@extra@context @sizes}%
4297         {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
4298     }%
4299   }%
4300 }
```

\MT@get@font@and@size   Translate any asterisks and split off the size.

```
4301 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
4302   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
4303 }
```

```
4304 \MT@define@code@key{encoding}{cfg}
4305 \MT@define@code@key@family    {cfg}
4306 \MT@define@code@key{series}   {cfg}
4307 \MT@define@code@key{shape}    {cfg}
4308 \MT@define@code@key@size      {cfg}
4309 \MT@define@code@key@font      {cfg}
```

\MT@define@opt@key

```
4310 \def\MT@define@opt@key#1#2{%
4311   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
4312     \MT@xdef@n{MT@#1@c@\MT@curr@set@name @#2}{##1}}}%
4313 }
```

\MT@listname@count   The options in the optional first argument.

```
4314 \newcount\MT@listname@count
4315 \MT@map@clist@c\MT@features{%
```

Use file name and line number as the list name if the user didn't bother to invent
one – also check whether the name already exists (in case more than one unnamed
list is loaded in the same line, for example \AtBeginDocument).

```
4316   \define@key{MT@#1@c}{name}[]{%
4317     \MT@ifempty{##1}{%
4318       \MT@ifdefined@n@TF{MT@#1@c@\MT@curr@file/\the\inputlineno}{%
4319         \global\advance\MT@listname@count\@ne
4320         \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno
4321                                  (\number\MT@listname@count)}%
4322       }{%
4323         \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
4324       }%
4325     }{%
4326       \MT@edef@n{MT@#1@c@name}{##1}%
4327       \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
4328         \MT@warning{Redefining \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'}%
4329       }%
4330     }%
4331     \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
4332   }%
4333   \MT@define@opt@key{#1}{load}%
4334   \MT@define@opt@key{#1}{factor}%
4335   \MT@define@opt@key{#1}{preset}%
4336   \MT@define@opt@key{#1}{inputenc}%
```

Only one context is allowed. This might change in the future.

```
4337   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
4338 }
4339 ⟨/package⟩
```

Automatically enable font copying if we find a protrusion or expansion context.
After the preamble, check whether font copying is enabled. For older pdfTeX
versions, disallow. It also works with LuaTeX 0.30 or newer.

```
4340 ⟨*pdf-|lua-⟩
4341 ⟨pdf-⟩\MT@requires@pdftex7{
4342   \define@key{MT@ex@c}{context}[]{%
4343     \MT@ifempty{#1}\relax{%
4344       \MT@glet\MT@copy@font\MT@copy@font@
4345       \def\MT@extra@context{#1}%
4346     }%
4347   }
4348   \MT@addto@setup{%
```

```
4349    \define@key{MT@ex@c}{context}[]{%
4350      \ifx\MT@copy@font\MT@copy@font@
4351        \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
4352      \else
4353        \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
4354            Ignoring `context' key\on@line}%
4355          {Either move the settings inside the preamble,\MessageBreak
4356            or load the package with the `copyfonts' option.}%
4357      \fi
4358    }%
4359  }
```

Protrusion contexts *might* also work without copying the font, so we don't issue an error but only a warning. The problem is that pdfTeX only allows one set of protrusion factors for a given font within one paragraph (those that are in effect at the end of the paragraph will be in effect for the whole paragraph). When different fonts are loaded – like in the example with the footnote markers – we don't need to copy the fonts.

```
4360    \define@key{MT@pr@c}{context}[]{%
4361      \MT@ifempty{#1}\relax{%
4362        \MT@glet\MT@copy@font\MT@copy@font@
4363        \def\MT@extra@context{#1}%
4364      }%
4365    }
4366    \MT@addto@setup{%
4367      \define@key{MT@pr@c}{context}[]{%
4368        \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
4369        \ifx\MT@copy@font\MT@copy@font@\else
4370          \MT@warning@nl{If protrusion contexts don't work as expected,
4371            \MessageBreak load the package with the `copyfonts' option}%
4372        \fi
4373      }%
4374    }
4375  ⟨/pdf-|lua-⟩
4376  ⟨*pdf-⟩
4377  }{
4378    \define@key{MT@ex@c}{context}[]{%
4379      \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
4380          or later. Ignoring `context' key\on@line}%
4381        {Upgrade pdftex.}%
4382    }
4383  ⟨/pdf-⟩
4384  ⟨*pdf-|xe-⟩
4385    \define@key{MT@pr@c}{context}[]{%
4386      \MT@error{Protrusion contexts only work with pdftex
4387  ⟨pdf-⟩        1.40.4\MessageBreak or later.
4388  ⟨xe-⟩          \MessageBreak or luatex.
4389          Ignoring `context' key\on@line}%
4390  ⟨pdf-⟩      {Upgrade pdftex.}%
4391  ⟨xe-⟩      {Use pdftex or luatex.}%
4392    }
4393  ⟨/pdf-|xe-⟩
4394  ⟨pdf-⟩}
```

\MT@warn@nodim

```
4395  ⟨*package⟩
4396  \def\MT@warn@nodim#1{%
4397    \MT@warning{`\@tempa' is not a dimension.\MessageBreak
4398                Ignoring it and setting values relative to\MessageBreak #1}%
4399  }
```

Protrusion codes may be relative to character width, or to any dimension.

```
4400  \define@key{MT@pr@c}{unit}[character]{%
4401    \MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@empty
```

```
4402    \def\@tempa{#1}%
4403    \MT@ifstreq\@tempa{character}\relax{%
```

Test whether it's a dimension, but do not translate it into its final form here, since it may be font-specific.

```
4404      \MT@ifdimen\@tempa
4405        {\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@tempa}%
4406        {\MT@warn@nodim{character widths}}%
4407    }%
4408 }
4409 ⟨/package⟩
```

Tracking may only be relative to a dimension.

```
4410 ⟨*pdf-|lua-⟩
4411 \define@key{MT@tr@c}{unit}[1em]{%
4412    \MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@empty
4413    \def\@tempa{#1}%
4414    \MT@ifdimen\@tempa
4415      {\MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@tempa}%
4416      {\MT@warn@nodim{1em}%
4417       \MT@gdef@n{MT@tr@c@\MT@curr@set@name @unit}{1em}}%
4418 }
4419 ⟨/pdf-|lua-⟩
```

Spacing and kerning codes may additionally be relative to space dimensions.

```
4420 ⟨*pdf-⟩
4421 \MT@map@clist@n{sp,kn}{%
4422    \define@key{MT@#1@c}{unit}[space]{%
4423      \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@empty
4424      \def\@tempa{##1}%
4425      \MT@ifstreq\@tempa{character}\relax{%
4426        \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\m@ne
4427        \MT@ifstreq\@tempa{space}\relax{%
4428          \MT@ifdimen\@tempa
4429            {\MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@tempa}%
4430            {\MT@warn@nodim{width of space}}%
4431        }%
4432      }%
4433    }%
4434 }
4435 ⟨/pdf-⟩
```

The first argument to \SetExpansion accepts some more options.

```
4436 ⟨*pdf-|lua-⟩
4437 \MT@map@clist@n{stretch,shrink,step}{%
4438    \define@key{MT@ex@c}{#1}[]{%
4439      \MT@ifempty{##1}\relax{%
4440        \MT@ifint{##1}{%
```

A space terminates the number.

```
4441          \MT@gdef@n{MT@ex@c@\MT@curr@set@name @#1}{##1 }%
4442        }{%
4443          \MT@warning{%
4444            Value `##1' for option `#1' is not a number.\MessageBreak
4445            Ignoring it}%
4446        }%
4447      }%
4448    }%
4449 }
4450 \define@key{MT@ex@c}{auto}[true]{%
4451    \def\@tempa{#1}%
4452    \csname if\@tempa\endcsname
```

Don't use autoexpand for pdfTeX version older than 1.20.

```
4453 ⟨pdf-⟩      \MT@requires@pdftex4%
```

```
4454 ⟨lua-⟩    \MT@requires@luatex3\relax
4455       {\MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}}%
4456 ⟨pdf-⟩       {\MT@warning{pdftex too old for automatic font expansion}}%
4457    \else
4458 ⟨pdf-⟩     \MT@requires@pdftex4%
4459 ⟨*lua-⟩
4460     \MT@requires@luatex3{%
4461     \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
4462                 luatex}}%
4463 ⟨/lua-⟩
4464       {\MT@glet@nc{MT@ex@c@\MT@curr@set@name @auto}\@empty}%
4465 ⟨pdf-⟩      \relax
4466    \fi
4467 }
```

Tracking: Interword spacing and outer kerning. The variant with space just in case
\SetTracking is called inside an argument (e.g., to \IfFileExists).

```
4468 \MT@define@opt@key{tr}{spacing}
4469 \MT@define@opt@key{tr}{outerspacing}
4470 \MT@define@opt@key{tr}{outerkerning}
```

Which ligatures should be disabled?

```
4471 \define@key{MT@tr@c}{noligatures}[]%
4472   {\MT@xdef@n{MT@tr@c@\MT@curr@set@name @noligatures}{#1}}
4473 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
4474 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
4475 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}
4476 ⟨/pdf-|lua-⟩
```

1.3.6 Character inheritance

\DeclareCharacterInheritance This macro may be used in the configuration files to declare characters that should
inherit protrusion resp. expansion values from other characters. Thus, there is no
need to define all accented characters (e.g., \'a, \'a, \^a, \~a, \"a, \r{a}, \k{a},
\u{a}), which will make the configuration files look much nicer and easier to
maintain. If a single character of an inheritance list should have a different value,
one can simply override it.

\MT@inh@feat The optional argument may be used to restrict the list to some features,
\MT@extra@inputenc and to specify an input encoding.

```
4477 ⟨*package⟩
4478 \renewcommand*\DeclareCharacterInheritance[1][]{%
4479   \let\MT@extra@context\@empty
4480   \let\MT@extra@inputenc\@undefined
4481   \let\MT@inh@feat\@empty
4482   \setkeys{MT@inh@}{#1}%
4483   \MT@begin@catcodes
4484   \MT@set@inh@list
4485 }
```

\MT@set@inh@list No need to create an inheritance list for tracking.

```
4486 \def\MT@set@inh@list#1#2{%
4487  \MT@ifempty\MT@inh@feat{%
4488    \MT@map@clist@c\MT@features{\begingroup
4489      \MT@ifstreq{##1}{tr}\relax{\MT@declare@char@inh{##1}{#1}{#2}}%
4490    \endgroup}%
4491  }{%
4492    \MT@map@clist@c\MT@inh@feat{\begingroup
4493      \KV@@sp@def\@tempa{##1}%
4494      \MT@ifempty\@tempa\relax{%
4495        \edef\@tempa{\csname MT@rbba@\@tempa\endcsname}%
4496        \MT@ifstreq\@tempa{tr}\relax{%
```

```
4497              \MT@exp@one@n\MT@declare@char@inh{\@tempa}{#1}{#2}}}%
4498        \endgroup}%
4499    }%
4500    \MT@end@catcodes
4501 }
```

The keys for the optional argument.

```
4502 \MT@map@clist@c\MT@features@long{%
4503    \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}}
4504 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

\MT@declare@char@inh   The lists cannot be given a name by the user.

```
4505 \def\MT@declare@char@inh#1#2#3{%
4506    \MT@edef@n{MT@#1@inh@name}%
4507       {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
4508    \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
4509    \MT@ifdefined@c@T\MT@extra@inputenc{%
4510       \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
4511 ⟨debug⟩\MT@dinfo{1}{creating inheritance list `\@nameuse{MT@#1@inh@name}'}%
4512    \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
4513    \def\MT@permutelist{#1@inh}%
4514    \setkeys{MT@inh}{#2}%
4515    \MT@permute
4516 }
```

Parse the second argument. \DeclareCharacterInheritance may also be set up for various combinations. We can reuse the key setup from the configuration lists (\Set...).

```
4517 \MT@define@code@key{encoding}{inh}
4518 \MT@define@code@key@family    {inh}
4519 \MT@define@code@key{series}   {inh}
4520 \MT@define@code@key{shape}     {inh}
4521 \MT@define@code@key@size       {inh}
4522 \MT@define@code@key@font       {inh}
```

\MT@inh@do   Now parse the third argument, the inheritance lists. We define the commands \MT@inh@⟨name⟩@⟨slot⟩@, containing the inheriting characters. They will also be translated to slot numbers here, to save some time. The following will be executed only once, namely the first time this inheritance list is encountered (in \MT@set@⟨feature⟩@codes).

```
4523 \def\MT@inh@do#1,{%
4524    \ifx\relax#1\@empty \else
4525       \MT@inh@split #1==\relax
4526       \expandafter\MT@inh@do
4527    \fi
4528 }
```

\MT@inh@split   Only gather the inheriting characters here. Their codes will actually be set in \MT@set@⟨feature⟩@codes.

```
4529 ⟨/package⟩
4530 ⟨*pdf-|lua-|xe-⟩
4531 \def\MT@inh@split#1=#2=#3\relax{%
4532    \def\@tempa{#1}%
4533    \ifx\@tempa\@empty \else
4534       \expandafter\MT@has@inh@prefix\@tempa()\relax\@nil
4535       \MT@get@slot
4536 ⟨pdf-|lua-⟩    \ifnum\MT@char > \m@ne
4537 ⟨xe-⟩    \ifx\MT@char\@empty\else
4538       \let\MT@val\MT@char
4539       \MT@map@clist@n{#2}{%
4540          \def\@tempa{##1}%
4541          \ifx\@tempa\@empty \else
4542             \MT@get@slot
```

```
4543 ⟨pdf-|lua-⟩          \ifnum\MT@char > \m@ne
4544 ⟨xe-⟩            \ifx\MT@char\@empty\else
4545                \ifx\MT@inh@prefix\@empty
4546                  \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @\MT@val @}{{\MT@char}}%
4547                \else
4548                  \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @prefixes}%
4549                      {{{\MT@val}{\MT@char}\MT@inh@prefix@}}%
4550                \fi
4551              \fi
4552            \fi
4553          }%
4554 ⟨debug⟩\MT@dinfo@nl{2}{children of #1 (\MT@val):
4555 ⟨debug⟩     \@nameuse{MT@inh@\MT@listname @\ifx\MT@inh@prefix\@empty\MT@val @\else prefixes\fi}}%
4556      \fi
4557    \fi
4558 }
4559 ⟨/pdf-|lua-|xe-⟩
```

\MT@inh@prefix
\MT@has@inh@prefix

If the inheriting character is preceded by (⟨*prefix*⟩), where ⟨*prefix*⟩ is one of l, r or lr, this has a special meaning for protrusion. For the other features, we ignore these settings.

```
4560 ⟨*package⟩
4561 \def\MT@has@inh@prefix#1(#2)#3#4\@nil{%
4562    \let\MT@temp\relax
4563    \ifx\relax#3%
4564      \def\@tempa{#1#2}%
4565      \let\MT@inh@prefix\@empty
4566    \else
4567      \MT@ifstreq{\MT@feat}{pr}{%
4568        \MT@ifstreq{#2}{l}{\def\MT@inh@prefix@{{1000}{0}}}\@firstoftwo}{%
4569          \MT@ifstreq{#2}{r}{\def\MT@inh@prefix@{{0}{1000}}}\@firstoftwo}{%
4570            \MT@ifstreq{#2}{lr}{\def\MT@inh@prefix@{{500}{500}}}\@firstoftwo}{%
4571              \MT@warning@nl{`#2' is not a valid prefix in inheritance list%
4572                \MessageBreak\MT@listname. Ignoring it}%
4573              \@secondoftwo}}}%
4574        {\def\@tempa{#3}%
4575         \def\MT@inh@prefix{#2}%
4576         \@gobble}%
4577        {\@firstofone}%
4578      }{\@firstofone}%
4579      {\let\MT@char\m@ne
4580       \let\MT@temp\@gobble
4581      }%
4582    \fi
4583    \MT@temp
4584 }
```

1.3.7 Permutation

\MT@permute
\MT@permute@
\MT@permute@@
\MT@permute@@@
\MT@permute@@@@

Calling \MT@permute will define commands for all permutations of the specified font attributes of the form \MT@⟨*list type*⟩@/⟨*encoding*⟩/⟨*family*⟩/⟨*series*⟩/⟨*shape*⟩/⟨|*⟩ to be the expansion of \MT@⟨*list type*⟩@name, i.e., the name of the currently defined list. Size ranges are held in a separate macro called \MT@⟨*list type*⟩@/⟨*font axes*⟩@sizes, which in turn contains the respective ⟨*list name*⟩s attached to the ranges. So that,

```
\SetProtrusion
   { encoding = U,
     family   = {euroitc,euroitcs} }
   { E = {100,50} }
\SetProtrusion
   { encoding = U,
     family   = {euroitc,euroitcs},
     shape    = it* }
```

```
{ E = {100,} }
```

would yield the following assignments:

```
4585 \MT@gdef@n{MT@pr@c@U/euroitc///}{euroitc}
4586 \MT@gdef@n{MT@pr@c@U/euroitcs///}{euroitc}
4587 \MT@gdef@n{MT@pr@c@U/euroitc//it/}{euroitci}
4588 \MT@gdef@n{MT@pr@c@U/euroitcs//it/}{euroitci}
4589 \MT@gdef@n{MT@pr@c@euroitc}{E={100,50}}
4590 \MT@gdef@n{MT@pr@c@euroitci}{E={100,}}
4591 \def\MT@permute{%
4592   \let\MT@cnt@encoding\@ne
4593   \MT@permute@
```

Undefine commands for the next round.

```
4594   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
4595   \MT@glet\MT@tempsize\@undefined
4596 }
4597 \def\MT@permute@{%
4598   \let\MT@cnt@family\@ne
4599   \MT@permute@@
4600   \MT@increment\MT@cnt@encoding
4601   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
4602     \MT@permute@
4603 }
4604 \def\MT@permute@@{%
4605   \let\MT@cnt@series\@ne
4606   \MT@permute@@@
4607   \MT@increment\MT@cnt@family
4608   \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
4609     \MT@permute@@
4610 }
4611 \def\MT@permute@@@{%
4612   \let\MT@cnt@shape\@ne
4613   \MT@permute@@@@
4614   \MT@increment\MT@cnt@series
4615   \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
4616     \MT@permute@@@
4617 }
4618 \def\MT@permute@@@@{%
4619   \MT@permute@@@@@
4620   \MT@increment\MT@cnt@shape
4621   \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
4622     \MT@permute@@@@
4623 }
```

\MT@permute@@@@@   In order to save some memory, we can ignore unused encodings (inside the document).

```
4624 \def\MT@permute@@@@@{%
4625   \MT@permute@define{encoding}%
4626   \ifMT@document
4627     \ifx\MT@tempencoding\@empty \else
4628       \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
4629         {\expandafter\expandafter\expandafter\@gobble}%
4630     \fi
4631   \fi
4632   \MT@permute@@@@@
4633 }
```

\MT@permute@@@@@@

```
4634 \def\MT@permute@@@@@@{%
4635   \MT@permute@define{family}%
4636   \MT@permute@define{series}%
4637   \MT@permute@define{shape}%
4638   \edef\@tempa{\MT@tempencoding
4639             /\MT@tempfamily
```

```
4640                /\MT@tempseries
4641                /\MT@tempshape
4642                /\MT@ifdefined@c@T\MT@tempsize *}%
```

Some sanity checks: an encoding must be specified (unless nothing else is).

```
4643   \MT@ifstreq\@tempa{////}\relax{%
4644     \ifx\MT@tempencoding\@empty
4645       \MT@warning{%
4646         You have to specify an encoding for\MessageBreak
4647         \@nameuse{MT@abbr@\MT@permutelist} list
4648         `\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
4649         Ignoring it}%
4650     \else
4651       \MT@ifdefined@c@TF\MT@tempsize{%
```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```
4652         \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
4653           \MT@map@tlist@c\MT@tempsize\MT@check@rlist
4654         }%
4655         \MT@exp@cs\MT@xaddb
4656           {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
4657           \MT@tempsize
4658 ⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa,\MessageBreak
4659 ⟨debug⟩        sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
4660 ⟨debug⟩                      @sizes\endcsname}%
4661       }{%
```

Only one list can apply to a given combination. But we don't warn if the overridden list is to be loaded by the current one.

```
4662         \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
4663           \MT@ifstreq{\csname MT@\MT@permutelist @\@tempa\MT@extra@context\endcsname}%
4664             {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname @load\endcsname}%
4665             \relax{%
4666           \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
4667             `\@nameuse{MT@\MT@permutelist @name}' will\MessageBreak override
4668             list `\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
4669             for \MessageBreak font `\@tempa'}%
4670           }%
4671         }%
4672 ⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa
4673 ⟨debug⟩                \ifx\MT@extra@context\@empty\else\MessageBreak
4674 ⟨debug⟩                  (context: \MT@extra@context)\fi}%
4675       }%
4676       \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
4677         {\csname MT@\MT@permutelist @name\endcsname}%
4678     \fi
4679   }%
4680 }
```

\MT@permute@define   Define the commands.

```
4681 \def\MT@permute@define#1{%
4682   \@tempcnta=\csname MT@cnt@#1\endcsname\relax
4683   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
4684     {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
4685     {\MT@let@nc{MT@temp#1}\@empty}%
4686 }
```

\MT@permute@reset   Reset the commands.

```
4687 \def\MT@permute@reset#1{%
4688   \@tempcnta=\@ne
4689   \MT@loop
4690     \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
4691     \advance\@tempcnta\@ne
```

```
4692        \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
4693          \iftrue
4694          \iffalse
4695      \MT@repeat
4696  }
```

`\MT@check@rlist`   For every new range item in `\MT@tempsize`, check whether it overlaps with ranges in the existing list.

```
4697  \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

`\MT@check@rlist@`   Define the current new range and . . .

```
4698  \def\MT@check@rlist@#1#2#3{%
4699    \def\@tempb{#1}%
4700    \def\@tempc{#2}%
4701    \MT@if@false
4702    \MT@exp@cs\MT@map@tlist@c
4703      {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
4704      \MT@check@range
4705  }
```

`\MT@check@range`   . . . recurse through the list of existing ranges.

```
4706  \def\MT@check@range#1{\expandafter\MT@check@range@ #1}
```

`\MT@check@range@`   `\@tempb` and `\@tempc` are lower resp. upper bound of the new range, ⟨#1⟩ and ⟨#2⟩ those of the existing range. ⟨#3⟩ is the list name.

```
4707  \def\MT@check@range@#1#2#3{%
4708    \MT@ifdim{#2}=\m@ne{%
4709      \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
4710        \MT@ifdim\@tempb={#1}\MT@if@true\relax
4711      }{%
```

- Item in list is a simple size, new item is a range.

```
4712        \MT@ifdim\@tempb>{#1}\relax{%
4713          \MT@ifdim\@tempc>{#1}{%
4714            \MT@if@true
4715            \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
4716          }\relax
4717        }%
4718      }%
4719    }{%
4720      \MT@ifdim\@tempc=\m@ne{%
```

- Item in list is a range, new item is a simple size.

```
4721        \MT@ifdim\@tempb<{#2}{%
4722          \MT@ifdim\@tempb<{#1}\relax\MT@if@true
4723        }\relax
4724      }{%
```

- Both items are ranges.

```
4725        \MT@ifdim\@tempb<{#2}{%
4726          \MT@ifdim\@tempc>{#1}{%
4727            \MT@if@true
4728            \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
4729          }\relax
4730        }\relax
4731      }%
4732    }%
4733    \ifMT@if@
4734      \MT@ifstreq{#3}%
```

```
4735        {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname}@load\endcsname}%
4736        \relax{%
4737      \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
4738        `\@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
4739        list `#3' for font \@tempa,\MessageBreak size \@tempb}%
4740      }%
```

If we've already found a conflict with this item, we can skip the rest of the list.

```
4741      \expandafter\MT@tlist@break
4742    \fi
4743 }
```

1.4 Package options

1.4.1 Declaring the options

`\ifMT@opt@expansion`
`\ifMT@opt@auto`
`\ifMT@opt@DVI`

Keep track of whether the user explicitly set these options.

```
4744 \newif\ifMT@opt@expansion
4745 \newif\ifMT@opt@auto
4746 \newif\ifMT@opt@DVI
```

`\MT@optwarn@admissible`

Some warnings.

```
4747 \def\MT@optwarn@admissible#1#2{%
4748   \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
4749                 `#2'. Assuming `false'}%
4750 }
```

`\MT@optwarn@nan`

```
4751 ⟨/package⟩
4752 ⟨*package|letterspace⟩
4753 ⟨plain⟩\MT@requires@latex1{
4754 \def\MT@optwarn@nan#1#2{%
4755   \MT@warning@nl{Value `#1' for option `#2' is not a\MessageBreak number.
4756                 Using default value of \number\@nameuse{MT@#2@default}}%
4757 }
4758 ⟨plain⟩}\relax
4759 ⟨/package|letterspace⟩
4760 ⟨*package⟩
```

`\MT@opt@def@set`

```
4761 \def\MT@opt@def@set#1{%
4762   \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
4763     \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
4764   }{%
4765     \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
4766     \MT@warning@nl{The #1 set `\MT@val' is undeclared.\MessageBreak
4767                   Using set `\@nameuse{MT@\@tempb @setname}' instead}%
4768   }%
4769 }
```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a ⟨set name⟩.

```
4770 \MT@map@clist@n{protrusion,expansion}{%
4771   \define@key{MT}{#1}[true]{%
4772     \csname MT@opt@#1true\endcsname
4773     \MT@map@clist@n{##1}{%
4774       \KV@@sp@def\MT@val{####1}%
4775       \MT@ifempty\MT@val\relax{%
4776         \csname MT@#1true\endcsname
4777         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4778         \MT@ifstreq\MT@val{true}\relax
4779         {%
```

```
4780              \MT@ifstreq\MT@val{false}{%
4781                \csname MT@#1false\endcsname
4782              }{%
4783                \MT@ifstreq\MT@val{compatibility}{%
4784                  \MT@let@nc{MT@\@tempb @level}\@ne
4785                }{%
4786                  \MT@ifstreq\MT@val{nocompatibility}{%
4787                    \MT@let@nc{MT@\@tempb @level}\tw@
4788                  }{%
```

If everything failed, it should be a set name.

```
4789                    \MT@opt@def@set{#1}%
4790                  }%
4791                }%
4792              }%
4793            }%
4794          }%
4795        }%
4796    }%
4797 }
```

`activate` is a shortcut for `protrusion` and `expansion`.

```
4798 \define@key{MT}{activate}[true]{%
4799    \setkeys{MT}{protrusion={#1}}%
4800    \setkeys{MT}{expansion={#1}}%
4801 }
```

`spacing`, `kerning` and `tracking` do not have a compatibility level.

```
4802 \MT@map@clist@n{spacing,kerning,tracking}{%
4803    \define@key{MT}{#1}[true]{%
4804      \MT@map@clist@n{##1}{%
4805        \KV@@sp@def\MT@val{####1}%
4806        \MT@ifempty\MT@val\relax{%
4807          \csname MT@#1true\endcsname
4808          \MT@ifstreq\MT@val{true}\relax
4809          {%
4810            \MT@ifstreq\MT@val{false}{%
4811              \csname MT@#1false\endcsname
4812            }{%
4813              \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4814              \MT@opt@def@set{#1}%
4815            }%
4816          }%
4817        }%
4818      }%
4819    }%
4820 }
```

\MT@def@bool@opt   The `true`/`false` options: `draft` (may be inherited from the class options), `auto`, `selected`, `babel`, `DVIoutput`, `defersetup`, `copyfonts`.

```
4821 \def\MT@def@bool@opt#1#2{%
4822    \define@key{MT}{#1}[true]{%
4823      \def\@tempa{##1}%
4824      \MT@ifstreq\@tempa{true}\relax{%
4825        \MT@ifstreq\@tempa{false}\relax{%
4826          \MT@optwarn@admissible{##1}{#1}%
4827          \def\@tempa{false}%
4828        }%
4829      }%
4830      #2%
4831    }%
4832 }
```

Boolean options that only set the switch.

```
4833 \MT@map@clist@n{draft,selected,babel}{%
```

```
4834    \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
4835  \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotrue}
```

The `DVIoutput` option will change `\pdfoutput` immediately to minimise the risk of confusing other packages.

```
4836  ⟨/package⟩
4837  ⟨*pdf-|lua-|xe-⟩
4838  ⟨lua-⟩\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
4839  \MT@def@bool@opt{DVIoutput}{%
4840    \csname if\@tempa\endcsname
4841  ⟨*pdf-|lua-⟩
4842      \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
4843      \pdfoutput\z@
4844    \else
4845      \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
4846      \pdfoutput\@ne
4847  ⟨/pdf-|lua-⟩
4848  ⟨xe-⟩     \MT@warning@nl{Ignoring `DVIoutput' option}%
4849    \fi
4850  }
4851  ⟨/pdf-|lua-|xe-⟩
```

Setting the `defersetup` option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is *undocumented*, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

```
4852  ⟨*package⟩
4853  \MT@def@bool@opt{defersetup}{%
4854    \csname if\@tempa\endcsname \else
4855      \AtEndOfPackage{%
4856        \MT@setup@
4857        \let\MT@setup@\@empty
4858        \let\MT@addto@setup\@firstofone
4859      }%
4860    \fi
4861  }
4862  ⟨/package⟩
```

`copyfonts` will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This options is also *undocumented* in the hope that we can always find out automatically whether it's required. It also works with LuaTeX 0.30 or newer.

```
4863  ⟨*pdf-|lua-⟩
4864  ⟨pdf-⟩\MT@requires@pdftex7{
4865    \MT@def@bool@opt{copyfonts}{%
4866      \csname if\@tempa\endcsname
4867        \MT@glet\MT@copy@font\MT@copy@font@
4868      \else
4869        \MT@glet\MT@copy@font\relax
4870      \fi
4871    }
4872  ⟨pdf-⟩}{
4873  ⟨/pdf-|lua-⟩
4874  ⟨*pdf-|xe-⟩
4875    \MT@def@bool@opt{copyfonts}{%
4876      \csname if\@tempa\endcsname
4877        \MT@error
4878  ⟨pdf-⟩        {The pdftex version you are using is too old\MessageBreak
4879  ⟨pdf-⟩         to use the `copyfonts' option}{Upgrade pdftex.}%
4880  ⟨xe-⟩         {The `copyfonts' option does not work with xetex}
```

```
4881 ⟨xe-⟩          {Use pdftex or luatex instead.}%
4882    \fi
4883  }
4884 ⟨pdf-⟩}
4885 ⟨/pdf-|xe-⟩
```

final is the opposite to draft. It's only kept for backwards compatibility.

```
4886 ⟨*package⟩
4887 \MT@def@bool@opt{final}{}
```

The disable option replaces the draft option, which could be inherited from the class options. The third value ifdraft mimicks this behaviour.

```
4888 \define@key{MT}{disable}[true]{%
4889   \def\@tempa{#1}%
4890   \MT@ifstreq\@tempa{true}\MT@disabletrue{%
4891     \MT@ifstreq\@tempa{ifdraft}{\ifMT@draft\MT@disabletrue\fi}{%
4892       \MT@ifstreq\@tempa{false}\relax{%
4893         \MT@optwarn@admissible{#1}{disable}%
4894       }%
4895     }%
4896   }%
4897 }
```

For verbose output, we redefine \MT@vinfo.

```
4898 \define@key{MT}{verbose}[true]{%
4899   \let\MT@vinfo\MT@info@nl
4900   \def\@tempa{#1}%
4901   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
4902     \MT@ifstreq\@tempa{errors}{%
4903       \let\MT@warning    \MT@warn@err
4904       \let\MT@warning@nl\MT@warn@err
4905     }{%
4906       \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
4907       \MT@ifstreq\@tempa{silent}{%
4908         \let\MT@warning    \MT@info
4909         \let\MT@warning@nl\MT@info@nl
4910       }{%
4911         \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
4912       }%
4913     }%
4914   }%
4915 }
4916 ⟨/package⟩
```

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```
4917 ⟨*package|letterspace⟩
4918 ⟨plain⟩\MT@requires@latex1{
4919 \MT@map@clist@n{%
4920 ⟨package⟩    stretch,shrink,step,%
4921    letterspace}{%
4922 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
4923   \def\@tempa{##1 }%
```

No nonsense in \MT@factor et al.? A space terminates the number.

```
4924   \MT@ifint\@tempa
4925     {\MT@edef@n{MT@#1}{\@tempa}}%
4926     {\MT@optwarn@nan{##1}{#1}}%
4927 }%
4928 }
4929 ⟨plain⟩}\relax
4930 ⟨/package|letterspace⟩
```

factor will define the protrusion factor only.

```
4931 ⟨*package⟩
4932 \define@key{MT}{factor}[\MT@factor@default]{%
4933   \def\@tempa{#1 }%
4934   \MT@ifint\@tempa
4935     {\edef\MT@pr@factor{\@tempa}}
4936     {\MT@optwarn@nan{#1}{factor}}%
4937 }
```

Unit for protrusion codes.

```
4938 \define@key{MT}{unit}[character]{%
4939   \def\@tempa{#1}%
4940   \MT@ifstreq\@tempa{character}\relax{%
4941     \MT@ifdimen\@tempa
4942       {\let\MT@pr@unit\@tempa}%
4943       {\MT@warning@nl{`\@tempa' is not a dimension.\MessageBreak
4944               Ignoring it and setting values relative to\MessageBreak
4945               character widths}}%
4946   }%
4947 }
```

\MT@patches@list
\MT@nopatches@list

The patch and nopatch options. Remember chosen option for later (\relax means 'all', \@empty means 'none').

```
4948 \let\MT@patches@list\relax
4949 \let\MT@nopatches@list\@empty

4950 \define@key{MT}{patch}[all]{%
4951   \def\@tempa{#1}%
4952   \MT@ifstreq\@tempa{all}
4953     \relax
4954     {\MT@ifstreq\@tempa{none}
4955       {\let\MT@patches@list\@empty}
4956       {\def\MT@patches@list{#1}}}%
4957 }
4958 \define@key{MT}{nopatch}[all]{%
4959   \def\@tempa{#1}%
4960   \MT@ifstreq\@tempa{all}
4961     {\let\MT@nopatches@list\relax}
4962     {\MT@ifstreq\@tempa{none}
4963       \relax
4964       {\def\MT@nopatches@list{#1}}}%
4965 }
```

We can only apply the patches AtBeginDocument.

```
4966 \MT@addto@setup{%
4967   \ifx\MT@patches@list\relax
4968     \let\MT@patches@list\MT@patches@def
4969   \fi
4970   \ifx\MT@nopatches@list\@empty\else
4971     \ifx\MT@nopatches@list\relax
4972       \let\MT@nopatches@list\MT@patches@def
4973     \fi
4974     \MT@map@clist@c\MT@nopatches@list{%
4975       \MT@rem@from@clist{#1}\MT@patches@list}%
4976   \fi
4977   \ifx\MT@patches@list\@empty\else
4978 ^^X    \MT@map@clist@c\MT@patches@list{\MT@apply@patch{#1}}%
4979 ^^Q    \MT@warning@nl{Patches require the etex extensions. Ignoring them}%
4980   \fi
4981 }
```

1.4.2 Loading the definition file

Load the engine-specific code (as strewn across this file).

```
4982 \input{microtype-\MT@engine tex.def}
```

1.4.3   Reading the configuration file

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as modern TeX systems have switched to the pdfTeX engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfTeX.)

```
4983 \MT@protrusiontrue
4984 ⟨/package⟩
4985 ⟨*pdf-|lua-⟩
4986 \ifnum\pdfoutput<\@ne \else
```

Also, we only enable expansion by default if pdfTeX can expand the fonts automatically.

```
4987 ⟨pdf-⟩   \MT@requires@pdftex4{
4988         \MT@expansiontrue
4989 ⟨pdf-⟩     \MT@autotrue
4990 ⟨pdf-⟩   }\relax
4991 \fi
4992 ⟨lua-⟩\MT@autotrue
4993 ⟨/pdf-|lua-⟩
```

\MT@config@file
\MT@get@config

The main configuration file will be loaded before processing the package options. However, the `config` option must of course be evaluated beforehand. We also have to define a no-op for the regular option processing later.

```
4994 ⟨*package⟩
4995 \define@key{MT}{config}[]{\relax}
4996 \def\MT@get@config#1config=#2,#3\@nil{%
4997   \MT@ifempty{#2}%
4998     {\def\MT@config@file{\MT@MT.cfg}}%
4999     {\def\MT@config@file{#2.cfg}}%
5000 }
5001 \expandafter\expandafter\expandafter\MT@get@config
5002   \csname opt@\@currname.\@currext\endcsname,config=,\@nil
```

Load the file.

```
5003 \IfFileExists{\MT@config@file}{%
5004   \MT@info@nl{Loading configuration file \MT@config@file}%
5005   \MT@begin@catcodes
5006     \let\MT@begin@catcodes\relax
5007     \let\MT@end@catcodes\relax
5008     \let\MT@curr@file\MT@config@file
5009     \input{\MT@config@file}%
5010   \endgroup
5011 }{\MT@warning@nl{%
5012     Could not find configuration file `\MT@config@file'!\MessageBreak
5013     This will almost certainly cause undesired results.\MessageBreak
5014     Please fix your installation}%
5015 }
```

\MT@check@active@set

We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by \DeclareMicrotypeSetDefault (this is done at the end of the preamble).

```
5016 \def\MT@check@active@set#1{%
5017   \MT@ifdefined@n@TF{MT@#1@setname}{%
5018     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
5019   }{%
5020     \MT@ifdefined@n@TF{MT@default@#1@set}{%
5021       \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
```

```
5022        \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
5023      }{%
```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '@', and issue a warning.

```
5024        \MT@gdef@n{MT@#1@setname}{@}%
5025        \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
5026                      \MessageBreak Using empty set}%
5027      }%
5028    }%
5029 }
```

1.4.4 Hook for other packages

\Microtype@Hook   This hook may be used by font package authors, e.g., to declare alias fonts. If it is defined, it will be executed here, i.e., after the main configuration file has been loaded, and before the package options are evaluated.

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using \@ifpackageloaded in the font package was not viable), and (2) checking \AtBeginDocument could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren't overwritten. Example:

```
\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\@ifpackageloaded{microtype}
  \MinionPro@MT@Hook
  {\@ifundefined{Microtype@Hook}
    {\let\Microtype@Hook\MinionPro@MT@Hook}
    {\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}}
```

\MicroType@Hook with a capital T (which only existed in version 1.7) is now officially deprecated.

```
5030 \MT@ifdefined@c@T\MicroType@Hook{\MT@error{%
5031   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
5032   Use \string\Microtype@Hook\space instead}
5033   {You might want to inform the font package authors.}\MicroType@Hook}
5034 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

1.4.5 Changing options later

\microtypesetup   Inside the preamble, \microtypesetup accepts the same options as the package (un-
\MT@define@optionX   less defersetup=false). In the document body, it accepts the options: protrusion, expansion, activate, tracking, spacing and kerning (but specifying font sets is not allowed), and patch and nopatch.

```
5035 \def\microtypesetup{\setkeys{MT}}
5036 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
5037 ⟨/package⟩
5038 ⟨*pdf-|lua-|xe-⟩
5039 \def\MT@define@optionX#1#2{%
5040   \define@key{MTX}{#1}[true]{%
5041     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
5042     \MT@map@clist@n{##1}{%
```

```
5043        \KV@@sp@def\MT@val{####1}%
5044        \MT@ifempty\MT@val\relax{%
5045          \@tempcnta=\m@ne
5046          \MT@ifstreq\MT@val{true}{%
```

Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

```
5047            \MT@checksetup{#1}{%
5048              \@tempcnta=\csname MT@\@tempb @level\endcsname
5049              \MT@vinfo{Enabling #1
5050                    (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
5051            }%
5052          }{%
5053            \MT@ifstreq\MT@val{false}{%
5054              \@tempcnta=\z@
5055              \MT@vinfo{Disabling #1\on@line}%
5056            }{%
5057              \MT@ifstreq\MT@val{compatibility}{%
5058                \MT@checksetup{#1}{%
5059                  \@tempcnta=\@ne
5060                  \MT@let@nc{MT@\@tempb @level}\@ne
5061                  \MT@vinfo{Setting #1 to level 1\on@line}%
5062                }%
5063              }{%
5064                \MT@ifstreq\MT@val{nocompatibility}{%
5065                  \MT@checksetup{#1}{%
5066                    \@tempcnta=\tw@
5067                    \MT@let@nc{MT@\@tempb @level}\tw@
5068                    \MT@vinfo{Setting #1 to level 2\on@line}%
5069                  }%
5070                }{\MT@error{Value `\MT@val' for key `#1' not recognised}
5071                        {Use any of `true', `false', `compatibility' or
5072                         `nocompatibility'.}%
5073                }%
5074              }%
5075            }%
5076          }%
5077          \ifnum\@tempcnta>\m@ne
5078            #2\@tempcnta\relax
5079          \fi
5080        }%
5081      }%
5082    }%
5083 }
```

\MT@checksetup   Test whether the feature wasn't disabled in the package options.

```
5084 \def\MT@checksetup#1{%
5085   \csname ifMT@#1\endcsname
5086     \expandafter\@firstofone
5087   \else
5088     \MT@error{You cannot enable #1 if it was disabled\MessageBreak
5089             in the package options}{Load microtype with #1 enabled.}%
5090     \expandafter\@gobble
5091   \fi
5092 }
5093 \MT@define@optionX{protrusion}\MT@protrudechars
5094 ⟨/pdf-|lua-|xe-⟩
5095 ⟨*pdf-|lua-⟩
5096 \MT@define@optionX{expansion}\MT@adjustspacing
```

\MT@protrudechars

\MT@adjustspacing
```
5097 ⟨*lua-⟩
5098 \MT@requires@luatex4{
```

```
5099      \let\pdfprotrudechars\protrudechars
5100      \let\pdfadjustspacing\adjustspacing
5101  }\relax
```
5102 ⟨/lua-⟩
```
5103  \let\MT@protrudechars\pdfprotrudechars
5104  \let\MT@adjustspacing\pdfadjustspacing
```
5105 ⟨/pdf-|lua-⟩
5106 ⟨*xe-⟩
```
5107  \let\MT@protrudechars\XeTeXprotrudechars
5108  \define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
```
5109 ⟨/xe-⟩

\MT@define@optionX@   The same for `tracking`, `spacing` and `kerning`, which do not have a `compatibility` level.

5110 ⟨*pdf-|lua-⟩
5111 ⟨pdf-⟩\MT@requires@pdftex6{
5112 ⟨lua-⟩\MT@requires@luatex3{
```
5113    \def\MT@define@optionX@#1#2{%
5114      \define@key{MTX}{#1}[true]{%
5115        \MT@map@clist@n{##1}{%
5116          \KV@@sp@def\MT@val{####1}%
5117          \MT@ifempty\MT@val\relax{%
5118            \@tempcnta=\m@ne
5119            \MT@ifstreq\MT@val{true}{%
5120              \MT@checksetup{#1}{%
5121                \@tempcnta=\@ne
5122                \MT@vinfo{Enabling #1\on@line}%
5123              }%
5124            }{%
5125              \MT@ifstreq\MT@val{false}{%
5126                \@tempcnta=\z@
5127                \MT@vinfo{Disabling #1\on@line}%
5128              }{\MT@error{Value `\MT@val' for key `#1' not recognised}
5129                       {Use either `true' or `false'}%
5130              }%
5131            }%
5132            \ifnum\@tempcnta>\m@ne
5133              #2\relax
5134            \fi
5135          }%
5136        }%
5137      }%
5138  }
```

We cannot simply let `\MT@tracking` relax, since this may select the already letter-spaced font instance.

```
5139    \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
5140                             \else \let\MT@tracking\MT@tracking@ \fi}
```
5141 ⟨pdf-⟩   `\MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}`
5142 ⟨pdf-⟩   `\MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta`
5143 ⟨pdf-⟩   `\pdfappendkern\@tempcnta}`
```
5144  }{
```
5145 ⟨/pdf-|lua-⟩
5146 ⟨*pdf-|lua-|xe-⟩

Disable for older pdfTEX versions and for XƎTEX and LuaTEX.

```
5147  \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
```
5148 ⟨lua-⟩}
```
5149  \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
5150  \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
```
5151 ⟨pdf-⟩}
```
5152  \define@key{MTX}{activate}[true]{%
5153    \setkeys{MTX}{protrusion={#1}}%
```
5154 ⟨pdf-|lua-⟩   `\setkeys{MTX}{expansion={#1}}}%`
```
5155  }
```

5156 ⟨/pdf-|lua-|xe-⟩

\MT@saved@setupfont   Disable everything – may be used as a temporary work-around in case setting up
                           fonts doesn't work under certain circumstances, but only until that specific problem
                           is fixed. These options are *undocumented*, as they completely deprive us of the
                           possibility to act – we're blind and paralysed.

5157 ⟨*package⟩
5158 \let\MT@saved@setupfont\MT@setupfont

5159 \define@key{MTX}{deactivate}[]{%
5160   \MT@info{Deactivate `\MT@MT' package}%
5161   \let\MT@setupfont\relax
5162 }
5163 \define@key{MTX}{reactivate}[]{%
5164   \MT@info{Reactivate `\MT@MT' package}%
5165   \let\MT@setupfont\MT@saved@setupfont
5166 }

      Apply or revert patches.

5167 \define@key{MTX}{patch}[all]{%
5168   \def\@tempa{#1}%
5169   \MT@ifstreq\@tempa{all}
5170   {\let\@tempa\MT@patches@def}
5171   {\MT@ifstreq\@tempa{none}
5172   {\let\@tempa\@empty}
5173   \relax}%
5174   \ifx\@tempa\@empty\else
5175 ^^X   \MT@map@clist@c\@tempa{\MT@apply@patch{##1}}%
5176 ^^Q   \MT@warning@nl{Patches require the etex extensions. Ignoring them}%
5177   \fi
5178 }
5179 \define@key{MTX}{nopatch}[all]{%
5180   \def\@tempa{#1}%
5181   \MT@ifstreq\@tempa{all}
5182   {\let\@tempa\MT@patches@def}
5183   {\MT@ifstreq\@tempa{none}
5184   {\let\@tempa\@empty}
5185   \relax}%
5186   \ifx\@tempa\@empty\else
5187 ^^X   \MT@map@clist@c\@tempa{\MT@undo@patch{##1}}%
5188   \fi
5189 }
5190 ⟨/package⟩

1.4.6 Processing the options

\MT@ProcessOptionsWithKV   Parse options.

5191 ⟨*package|letterspace⟩
5192 ⟨plain⟩\MT@requires@latex1{
5193 \def\MT@ProcessOptionsWithKV#1{%
5194   \let\@tempc\relax
5195   \let\MT@temp\@empty
5196 ⟨plain⟩ \MT@requires@latex2{
5197   \MT@map@clist@c\@classoptionslist{%
5198   \def\CurrentOption{##1}%
5199   \MT@ifdefined@n@T{KV@#1@\expandafter\MT@getkey\CurrentOption=\@nil}{%
5200   \edef\MT@temp{\MT@temp,\CurrentOption,}%
5201   \@expandtwoargs\@removeelement\CurrentOption
5202   \@unusedoptionlist\@unusedoptionlist
5203   }%
5204   }%
5205   \edef\MT@temp{\noexpand\setkeys{#1}%
5206   {\MT@temp\@ptionlist{\@currname.\@currext}}}%

eplain can handle package options.

```
5207 ⟨*plain⟩
5208   }{\edef\MT@temp{\noexpand\setkeys{#1}%
5209                   {\csname usepkg@options@\usepkg@pkg\endcsname}}}
5210 ⟨/plain⟩
5211   \MT@temp
5212   \MT@clear@options
5213 }
```

\MT@getkey   For key=val in class options.

```
5214 \def\MT@getkey#1=#2\@nil{#1}
```

```
5215 \MT@ProcessOptionsWithKV{MT}
5216 ⟨plain⟩}\relax
5217 ⟨/package|letterspace⟩
5218 ⟨*package⟩
```

Now we can take the appropriate actions. We also tell the log file which options
the user has chosen (in case it's interested).

```
5219 \MT@addto@setup{%
5220 \ifMT@disable
```

We disable most of what we've just defined in the 5220 lines above if we are
running in disable (aka. draft) mode.

```
5221   \MT@warning@nl{The `disable' option is in effect.\MessageBreak
5222                 Disabling all micro-typographic extensions.\MessageBreak
5223                 This might lead to different line and page breaks}%
5224   \let\MT@setupfont\relax
5225   \renewcommand*\LoadMicrotypeFile[1]{}%
5226   \renewcommand*\microtypesetup[1]{}%
5227   \renewcommand*\microtypecontext[1]{}%
5228   \renewcommand*\lsstyle{}%
5229 \else
5230   \MT@setup@PDF
5231   \MT@setup@copies
```

Fix the font sets.

```
5232   \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
5233   \MT@setup@protrusion
5234   \MT@setup@expansion
5235   \MT@setup@tracking
5236   \MT@setup@warntracking
5237   \MT@setup@spacing
5238   \MT@setup@kerning
5239   \MT@setup@noligatures
5240 }
5241 ⟨/package⟩
```

\MT@setup@PDF   pdfTeX can create DVI output, too. However, both the DVI viewer and dvips need
to find actual fonts. Therefore, expansion will only work if the fonts for different
degrees of expansion are readily available.

Some packages depend on the value of \pdfoutput and will get confused if it is
changed after they have been loaded. These packages are, among others: color,
graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf.
Instead of testing for each package (that's not our job), we only say that it was
microtype that changed it. This must be sufficient!

```
5242 ⟨*pdf-|lua-⟩
5243 \def\MT@setup@PDF{%
5244   \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
5245                \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
5246 }
```

\MT@setup@copies   Working on font copies?

```
5247 \def\MT@setup@copies{%
5248   \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
5249 }
5250 ⟨/pdf-|lua-⟩
5251 ⟨*xe-⟩
5252 \let\MT@setup@PDF\relax
5253 \let\MT@setup@copies\relax
5254 ⟨/xe-⟩
```

\MT@setup@protrusion   Protrusion.

```
5255 ⟨*pdf-|lua-|xe-⟩
5256 \def\MT@setup@protrusion{%
5257   \ifMT@protrusion
5258     \edef\MT@active@features{\MT@active@features,pr}%
5259     \MT@protrudechars\MT@pr@level
5260     \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
5261       \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
5262         factor: \number\MT@pr@factor\fi
5263       \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
5264     \MT@check@active@set{pr}%
5265   \else
5266     \let\MT@protrusion\relax
5267     \MT@info@nl{No character protrusion}%
5268   \fi
5269 }
5270 ⟨/pdf-|lua-|xe-⟩
```

\MT@setup@expansion   For DVI output, the user must have explicitly passed the expansion option to
                          the package. Under LuaTEX, expansion works quite differently: the glyphs will
                          be positioned as if they were transformed, without actually being transformed.
                          Since this could still be considered a viable option, we don't disable the feature
                          completely, but issue a warning.

```
5271 ⟨*pdf-|lua-⟩
5272 \def\MT@setup@expansion{%
5273   \ifnum\pdfoutput<\@ne
5274     \ifMT@opt@expansion
5275 ⟨*lua-⟩
5276       \ifMT@expansion
5277         \MT@requires@luatex3{%
5278           \MT@warning@nl{Font expansion doesn't work properly with luatex in\MessageBreak
5279             DVI mode: the glyphs won't be actually transformed,\MessageBreak
5280             but will only be shifted. You might want to use\MessageBreak
5281             pdflatex instead. I'll continue anyway ..}%
5282         %\MT@expansionfalse
5283         }\relax
5284       \fi
5285 ⟨/lua-⟩
5286     \else
5287       \MT@expansionfalse
5288     \fi
5289   \fi
5290   \ifMT@expansion
```

Set up the values for font expansion: if stretch has not been specified, we take the
default value of 20.

```
5291     \ifnum\MT@stretch=\m@ne
5292       \let\MT@stretch\MT@stretch@default
5293     \fi
```

If shrink has not been specified, it will inherit the value from stretch.

```
5294     \ifnum\MT@shrink=\m@ne
5295       \let\MT@shrink\MT@stretch
```

```
5296        \fi
```

If step has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for microtype.pdf with step=1 compared to step=5). With older versions, we set it to min(stretch,shrink)/5, rounded off, minimum value 1.

```
5297       \ifnum\MT@step=\m@ne
5298 ⟨pdf-⟩    \MT@requires@pdftex6{%
5299          \def\MT@step{1 }%
5300 ⟨*pdf-⟩
5301       }{%
5302         \ifnum\MT@stretch>\MT@shrink
5303           \ifnum\MT@shrink=\z@
5304             \@tempcnta=\MT@stretch
5305           \else
5306             \@tempcnta=\MT@shrink
5307           \fi
5308         \else
5309           \ifnum\MT@stretch=\z@
5310             \@tempcnta=\MT@shrink
5311           \else
5312             \@tempcnta=\MT@stretch
5313           \fi
5314         \fi
5315         \divide\@tempcnta 5\relax
5316         \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
5317         \edef\MT@step{\number\@tempcnta\space}%
5318       }%
5319 ⟨/pdf-⟩
5320       \fi
5321       \ifnum\MT@step=\z@
5322         \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
5323             Setting it to one}%
5324         \def\MT@step{1 }%
5325       \fi
```

\MT@auto   Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the *hz* programme really usable. It must be either 'autoexpand' or empty (or '1000' for older versions of pdfTeX). With LuaTeX, we just leave it empty, as there's actually no difference – non-automatic font expansion doesn't work anymore. In LuaTeX 1.0.6, the 'autoexpand' option seems to have been removed altogether and would trigger a warning.

```
5326       \let\MT@auto\@empty
5327       \ifMT@auto
```

We turn off automatic expansion if output mode is DVI.

```
5328 ⟨*pdf-⟩
5329       \MT@requires@pdftex4{%
5330         \ifnum\pdfoutput<\@ne
5331           \ifMT@opt@auto
5332             \MT@error{%
5333               Automatic font expansion only works for PDF output.\MessageBreak
5334               However, you are creating a DVI file}
5335             {If you have created expanded fonts instances, remove `auto' from%
5336               \MessageBreak the package options. Otherwise, you have to switch
5337               off expansion\MessageBreak completely.}%
5338           \fi
5339           \MT@autofalse
5340         \else
5341           \def\MT@auto{autoexpand}%
5342         \fi
```

Also, if pdfTeX is too old.

```
5343          }{%
5344            \MT@error{%
5345              The pdftex version you are using is too old for\MessageBreak
5346              automatic font expansion}%
5347            {If you have created expanded fonts instances, remove `auto' from\MessageBreak
5348              the package options. Otherwise, you have to switch off expansion\MessageBreak
5349              completely, or upgrade pdftex to version 1.20 or newer.}%
5350            \MT@autofalse
5351            \def\MT@auto{1000 }%
5352          }%
5353 ⟨/pdf-⟩
5354 ⟨lua-⟩        \MT@requires@luatex3\relax{\def\MT@auto{autoexpand}}%
5355        \else
5356 ⟨*pdf-⟩
```

No automatic expansion.

```
5357          \MT@requires@pdftex4\relax{%
5358            \def\MT@auto{1000 }%
5359          }%
5360 ⟨/pdf-⟩
5361 ⟨*lua-⟩
5362          \MT@requires@luatex3{%
5363            \ifMT@opt@auto
5364              \MT@error{Non-automatic font expansion does not work with\MessageBreak
5365                      luatex}{Remove `auto=false' from the package options, or use pdftex.}%
5366              \MT@autotrue
5367            \fi
5368          }\relax
5369 ⟨/lua-⟩
5370        \fi
```

Choose the appropriate macro for selected expansion.

```
5371      \ifMT@selected
5372        \let\MT@set@ex@codes\MT@set@ex@codes@s
5373      \else
5374        \let\MT@set@ex@codes\MT@set@ex@codes@n
5375      \fi
```

Filter out stretch=0,shrink=0, since it would result in a pdfTeX error.

```
5376      \ifnum\MT@stretch=\z@
5377        \ifnum\MT@shrink=\z@
5378          \MT@warning@nl{%
5379            Both the stretch and shrink limit are set to zero.\MessageBreak
5380            Disabling font expansion}%
5381          \MT@expansionfalse
5382        \fi
5383      \fi
5384    \fi
5385    \ifMT@expansion
5386      \edef\MT@active@features{\MT@active@features,ex}%
5387      \MT@adjustspacing\MT@ex@level
5388      \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
5389              (level \number\MT@ex@level),\MessageBreak
5390              stretch: \number\MT@stretch, shrink: \number\MT@shrink,
5391              step: \number\MT@step, \ifMT@selected\else non-\fi selected}%
```

\MT@check@step   Check whether stretch and shrink are multiples of step.

```
5392    \def\MT@check@step##1{%
5393      \@tempcnta=\csname MT@##1\endcsname
5394      \divide\@tempcnta \MT@step
5395      \multiply\@tempcnta \MT@step
5396      \ifnum\@tempcnta=\csname MT@##1\endcsname\else
5397        \MT@warning@nl{The ##1 amount is not a multiple of step.\MessageBreak
5398                    The effective maximum ##1 is \the\@tempcnta\space
```

```
5399                        (step \number\MT@step)}%
5400        \fi
5401      }%
5402      \MT@check@step{stretch}%
5403      \MT@check@step{shrink}%
5404      \MT@check@active@set{ex}%
```

\showhyphens Inside \showhyphens, font expansion should be disabled. (Since 2017/01/10, the
LaTeX format contains a different version for X∃TEX, but since expansion doesn't
work with X∃TEX, we don't have to bother.) Since 2019/10/01, the command is
robust.

```
5405      \MT@ifdefined@n@TF{showhyphens }{%
5406        \def\MT@temp##1##2{%
5407          \expandafter\CheckCommand\csname showhyphens \endcsname[1]{##1}%
5408          \DeclareRobustCommand\showhyphens[1]{##2}}%
5409      }{%
5410        \def\MT@temp##1##2{%
5411          \CheckCommand*\showhyphens[1]{##1}%
5412          \gdef\showhyphens####1{##2}}%
5413      }%
5414      \MT@temp
5415        {\setbox0\vbox{\color@begingroup
5416          \everypar{}\parfillskip\z@skip
5417          \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
5418          \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}
5419        {\setbox0\vbox{\color@begingroup\pdfadjustspacing\z@
5420          \everypar{}\parfillskip\z@skip
5421          \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
5422          \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%
5423    \else
5424      \let\MT@expansion\relax
5425      \MT@info@nl{No font expansion}%
5426    \fi
5427  }
5428  ⟨/pdf-|lua-⟩
5429  ⟨*xe-⟩
5430  \def\MT@setup@expansion{%
5431    \ifMT@expansion
5432      \ifMT@opt@expansion
5433        \MT@error{Font expansion does not work with xetex}
5434                {Use pdftex or luatex instead.}%
5435      \fi
5436    \fi
5437  }
5438  ⟨/xe-⟩
```

\MT@setup@tracking Tracking, spacing and kerning.

```
5439  ⟨*pdf-|lua-⟩
5440  ⟨pdf-⟩\MT@requires@pdftex6{%
5441  ⟨lua-⟩\MT@requires@luatex3{%
5442    \def\MT@setup@tracking{%
5443      \ifMT@tracking
5444        \edef\MT@active@features{\MT@active@features,tr}%
5445        \MT@info@nl{Tracking enabled}%
5446        \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
5447        \ifMT@protrusion\else\MT@protrudechars\@ne\fi
5448      \else
5449        \let\MT@tracking\relax
5450        \MT@info@nl{No adjustment of tracking}%
5451      \fi
5452    }
5453  ⟨/pdf-|lua-⟩
```

\MT@setup@spacing

```
5454 ⟨*pdf-⟩
5455   \def\MT@setup@spacing{%
5456     \ifMT@spacing
5457       \edef\MT@active@features{\MT@active@features,sp}%
5458       \pdfadjustinterwordglue\@ne
5459       \MT@info@nl{Adjustment of interword spacing enabled}%
```

The ragged2e package sets interword spaces to a fixed value without glue. microtype's modifications can therefore have undesired effects. Therefore, we issue a warning.

```
5460       \MT@with@package@T{ragged2e}{%
5461         \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
5462           Adjustment of interword spacing may lead to\MessageBreak
5463           undesired results when used with `ragged2e'.\MessageBreak
5464           In this case, disable the `spacing' option}%
5465       }%
5466       \MT@check@active@set{sp}%
5467     \else
5468       \let\MT@spacing\relax
5469       \MT@info@nl{No adjustment of interword spacing}%
5470     \fi
5471   }
```

\MT@setup@spacing@check   Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue > 0. Why 1500? Because some packages redefine \frenchspacing.[5]

```
5472   \def\MT@setup@spacing@check{%
5473     \ifMT@spacing
5474       \ifMT@babel \else
5475         \ifnum\sfcode`\. > 1500
5476           \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
5477             \MT@warning@nl{%
5478               \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
5479               interword spacing will disable it. You might want\MessageBreak
5480               to add `\@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
5481               to your preamble}%
5482           }%
5483         \fi
5484       \fi
5485     \fi
5486   }
```

\MT@setup@kerning

```
5487   \def\MT@setup@kerning{%
5488     \ifMT@kerning
5489       \edef\MT@active@features{\MT@active@features,kn}%
5490       \pdfprependkern\@ne
5491       \pdfappendkern\@ne
5492       \MT@info@nl{Adjustment of character kerning enabled}%
5493       \MT@check@active@set{kn}%
5494     \else
5495       \let\MT@kerning\relax
5496       \MT@info@nl{No adjustment of character kerning}%
5497     \fi
5498   }
5499 ⟨/pdf-⟩
```

\MT@error@doesnt@work   If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error message. We also switch the features off for LuaTeX and X\kern-.125em\lower.5ex\hbox{E}\kern-.1667em TeX.

```
5500 ⟨pdf-|lua-⟩}{
5501 ⟨*lua-⟩
5502   \def\MT@setup@tracking{%
```

---

5   Cf. the c.t.t. thread '\frenchspacing with AMS packages and babel', started by Philipp Lehman on 16 August 2005, MID: ddtbajrob1@online.de

```
5503        \ifMT@tracking
5504          \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
5505            or newer. Switching it off}{Upgrade luatex.}%
5506          \MT@trackingfalse
5507          \MT@let@nc{MT@tracking}\relax
5508        \else
5509          \MT@info@nl{No adjustment of tracking (luatex too old)}%
5510        \fi
5511     }
5512   }
5513   ⟨/lua-⟩
5514   ⟨*pdf-|lua-|xe-⟩
5515     \def\MT@error@doesnt@work#1{%
5516       \csname ifMT@#1\endcsname
5517         \MT@error{The #1 feature only works with pdftex 1.40\MessageBreak
5518           or newer. Switching it off}
5519   ⟨pdf-⟩         {Upgrade pdftex.}%
5520   ⟨lua-|xe-⟩          {Use pdftex instead.}%
5521         \csname MT@#1false\endcsname
5522         \MT@let@nc{MT@#1}\relax
5523       \else
5524         \MT@info@nl{No adjustment of #1%
5525   ⟨pdf-⟩        \space(pdftex too old)%
5526         }%
5527       \fi
5528     }
5529   ⟨pdf-|xe-⟩  \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
5530     \def\MT@setup@kerning  {\MT@error@doesnt@work{kerning}}
5531     \def\MT@setup@spacing  {\MT@error@doesnt@work{spacing}}
5532   ⟨pdf-⟩}
5533   ⟨/pdf-|lua-|xe-⟩
```

\MT@setup@warntracking

```
5534   ⟨letterspace⟩\MT@addto@setup
5535   ⟨pdf-|lua-⟩\def\MT@setup@warntracking
```

\MT@warn@tracking@DVI   With pdfTeX, we issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```
5536   ⟨*pdf-|lua-|letterspace⟩
5537   {%
5538   ⟨*pdf-|letterspace⟩
5539     \ifnum\pdfoutput<\@ne
5540       \def\MT@warn@tracking@DVI{%
5541   ⟨letterspace⟩        \MT@pdf@or@lua{%
5542         \MT@warning@nl{%
5543           You are using tracking/letterspacing in DVI mode.\MessageBreak
5544           This will probably not work, unless the post-\MessageBreak
5545           processing program (dvips, dvipdfm(x), ...) is\MessageBreak
5546           able to create the virtual fonts on the fly}%
5547   ⟨letterspace⟩        }\relax
5548         \MT@glet\MT@warn@tracking@DVI\relax
5549       }%
5550     \else
5551   ⟨/pdf-|letterspace⟩
5552       \def\MT@warn@tracking@DVI{%
5553         \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
5554         \MT@glet\MT@warn@tracking@DVI\relax
5555       }%
5556   ⟨pdf-|letterspace⟩  \fi

5557     \ifnum\MT@letterspace=\m@ne
5558       \let\MT@letterspace\MT@letterspace@default
5559     \else
5560       \MT@ls@too@large\MT@letterspace
```

```
5561      \fi
5562   }
5563   ⟨/pdf-|lua-|letterspace⟩
5564   ⟨xe-⟩\let\MT@setup@warntracking\relax
```

\MT@setup@noligatures \DisableLigatures is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```
5565   ⟨*pdf-|lua-⟩
5566   \def\MT@setup@noligatures{%
5567   ⟨pdf-⟩   \MT@requires@pdftex5{%
5568            \ifMT@noligatures \else
5569              \let\MT@noligatures\relax
5570            \fi
5571   ⟨pdf-⟩   }\relax
5572   }
5573   ⟨/pdf-|lua-⟩
5574   ⟨xe-⟩\let\MT@setup@noligatures\relax
```

Remove the leading comma in \MT@active@features, and set the document switch to true.

```
5575   ⟨*package⟩
5576   \MT@addto@setup{%
5577     \ifx\MT@active@features\@empty \else
5578       \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
5579     \fi
5580     \MT@documenttrue
5581   }
```

\MT@set@babel@context Interaction with babel.

```
5582   \def\MT@set@babel@context#1{%
5583     \MT@ifdefined@n@TF{MT@babel@#1}{%
5584       \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
5585       \expandafter\MT@exp@one@n\expandafter\microtypecontext
5586         \csname MT@babel@#1\endcsname
5587     }{%
5588       \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
5589     }%
5590   }
```

\MT@shorthandoff Active characters can only be switched off if babel isn't loaded after microtype.

```
5591   \@ifpackageloaded{babel}{
5592     \def\MT@shorthandoff#1#2{%
5593       \MT@info@nl{Switching off #1 babel's active characters (#2)}%
5594       \shorthandoff{#2}}
5595   }{
5596     \def\MT@shorthandoff#1#2{%
5597       \MT@error{You must load `babel' before `\MT@MT'}
5598                {Otherwise, `\MT@MT' cannot switch off #1 babel's\MessageBreak
5599                 active characters.}}
5600   }
```

We patch the language switching commands to enable language-dependent setup.

```
5601   \MT@addto@setup{%
5602     \ifMT@babel
5603       \@ifpackageloaded{babel}{%
5604         \MT@info@nl{Redefining babel's language switching commands}%
5605         \let\MT@orig@select@language\select@language
5606         \def\select@language#1{%
5607           \MT@orig@select@language{#1}%
5608           \MT@set@babel@context{#1}%
5609         }%
5610         \let\MT@orig@foreign@language\foreign@language
5611         \def\foreign@language#1{%
5612           \MT@orig@foreign@language{#1}%
```

```
5613        \MT@set@babel@context{#1}%
5614      }%
5615      \ifMT@kerning
```

Disable French babel's active characters.

```
5616        \MT@if@false
5617        \MT@with@babel@and@T{french}  \MT@if@true
5618        \MT@with@babel@and@T{frenchb} \MT@if@true
5619        \MT@with@babel@and@T{francais}\MT@if@true
5620        \MT@with@babel@and@T{canadien}\MT@if@true
5621        \MT@with@babel@and@T{acadian} \MT@if@true
5622        \ifMT@if@\MT@shorthandoff{French}{:;!?}\fi
```

Disable Turkish babel's active characters.

```
5623        \MT@if@false
5624        \MT@with@babel@and@T{turkish} \MT@if@true
5625        \ifMT@if@\MT@shorthandoff{Turkish}{:!=}\fi
5626      \fi
```

In case babel was loaded before microtype:

```
5627        \MT@set@babel@context\languagename
5628      }{%
5629        \MT@warning@nl{You did not load the babel package.\MessageBreak
5630          The `babel' option won't have any effect}%
5631      }%
5632    \fi
5633 }
```

Now we close the \fi from \ifMT@disable.

```
5634 \MT@addto@setup{\fi
```

Set up the current font, most likely the normal font. This has to come after all of the setup (including anything from the preamble) has been dealt with.

```
5635    \selectfont}
```

\MT@curr@file   This is the current file (hopefully with the correct extension).

```
5636 \edef\MT@curr@file{\jobname.tex}
5637 ⟨/package⟩
```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```
5638 ⟨*package|letterspace⟩
5639 ⟨plain⟩\MT@requires@latex1{
5640 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
5641 ⟨plain⟩}\relax
5642 ⟨/package|letterspace⟩
```

Must come at the very, very end.

```
5643 ⟨package⟩\MT@ifdefined@c@T\MT@setup@spacing@check
5644 ⟨package⟩   {\AtBeginDocument{\MT@setup@spacing@check}}
```

Restore catcodes.

```
5645 ⟨package|letterspace⟩\MT@restore@catcodes
```

That was that.

2   Configuration files

Let's now write the font configuration files.

```
5646 ⟨*config⟩
5647
```

2.1   Font sets

We first declare some sets in the main configuration file.

```
5648 ⟨*m-t⟩
5649 %%% ----------------------------------------------------------------------
5650 %%% FONT SETS
5651
5652 \DeclareMicrotypeSet{all}
5653    { }
5654
5655 \DeclareMicrotypeSet{allmath}
5656    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }
5657
5658 \DeclareMicrotypeSet{alltext}
5659    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
5660
5661 \DeclareMicrotypeSet{allmath-nott}
5662    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
5663      family   = {rm*,sf*}
5664    }
5665
5666 \DeclareMicrotypeSet{alltext-nott}
5667    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
5668      family   = {rm*,sf*}
5669    }
5670
5671 \DeclareMicrotypeSet{basicmath}
5672    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
5673      family   = {rm*,sf*},
5674      series   = {md*},
5675      size     = {normalsize,footnotesize,small,large}
5676    }
5677
5678 \DeclareMicrotypeSet{basictext}
5679    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
5680      family   = {rm*,sf*},
5681      series   = {md*},
5682      size     = {normalsize,footnotesize,small,large}
5683    }
5684
5685 \DeclareMicrotypeSet{smallcaps}
5686    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
5687      shape    = {sc*,si,scit}
5688    }
5689
5690 \DeclareMicrotypeSet{footnotesize}
5691    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
5692      size     = {-small}
5693    }
5694
5695 \DeclareMicrotypeSet{scriptsize}
5696    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
```

```
5697      size     = {-footnotesize}
5698    }
5699
5700 \DeclareMicrotypeSet{normalfont}
5701    { font = */*/*/*/* }
5702
```

The default sets.

```
5703 %%% ------------------------------------------------------------------------
5704 %%% DEFAULT SETS
5705
5706 \DeclareMicrotypeSetDefault[protrusion]{alltext}
5707 \DeclareMicrotypeSetDefault[expansion] {alltext-nott}
5708 \DeclareMicrotypeSetDefault[spacing]   {alltext-nott}
5709 \DeclareMicrotypeSetDefault[kerning]   {alltext}
5710 \DeclareMicrotypeSetDefault[tracking]  {smallcaps}
5711
```

2.2 Font variants and aliases

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals): Additionally, we add the now common variants for Lining, Tabular, Oldstyle, and Tabular Oldstyle numbers.

```
5712 %%% ------------------------------------------------------------------------
5713 %%% FONT VARIANTS AND ALIASES
5714
5715 \DeclareMicrotypeVariants{x,j,w,a,d,0,1,-LF,-TLF,-OsF,-TOsF}
```

Other candidates: 2 (proportional digits), e (engraved), f (Fraktur), g (small text), h (shadow), l (outline), n (informal), p (ornaments), r (roman), s (sans serif), t (typewriter). I've omitted them since they seem hardly be used and/or they are actually more than just a variant, i.e., they shouldn't share a file.

Fonts that are 'the same': The fontspec package will set lmr as the default font, whose declarations for EU1/EU2/TU encoding are in mt-LatinModernRoman.cfg. Since 2016/12/03, the default encoding with X∃TEX and LuaTEX in the LATEX format is TU, even if fontspec is not loaded.

```
5716
5717 \MT@if@false
5718 \ifx\UnicodeEncodingName\@undefined\else
5719   \MT@ifstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
5720 \fi
5721 \ifMT@fontspec\MT@if@true\fi
5722 \ifMT@if@
5723 %% -- Computer/Latin Modern Roman
5724 \DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
5725   \else
5726 \DeclareMicrotypeAlias{lmr}{cmr}           % lmodern
5727 \fi
```

The Latin Modern fonts, the virtual fonts from the ae and zefonts and the eco and hfoldsty packages (oldstyle numerals), as well as mlmodern, all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later. We mustn't forget the Latin Modern math fonts.

```
5728 \DeclareMicrotypeAlias{lmsy}{cmsy}        %    "
5729 \DeclareMicrotypeAlias{lmm} {cmm}         %    "
5730 \DeclareMicrotypeAlias{aer} {cmr}         % ae
5731 \DeclareMicrotypeAlias{zer} {cmr}         % zefonts
```

```
5732 \DeclareMicrotypeAlias{cmor}{cmr}          % eco
5733 \DeclareMicrotypeAlias{hfor}{cmr}          % hfoldsty
5734 \DeclareMicrotypeAlias{mlmr}{cmr}          % mlmodern
5735 \DeclareMicrotypeAlias{mlmsy}{cmsy}        %     "
5736 \DeclareMicrotypeAlias{mlmm} {cmm}         %     "
```

Another, new Computer Modern extension. The `newcomputermodern` package loads it by file name.

```
5737 \DeclareMicrotypeAlias{NewCM10-Book.otf}   {New Computer Modern}
5738 \DeclareMicrotypeAlias{NewCM10-Regular.otf}{New Computer Modern}
```

CMU Serif can use the settings from New Computer Modern too.

```
5739 \DeclareMicrotypeAlias{CMU Serif}          {New Computer Modern}
```

The packages `pxfonts` and `txfonts` fonts inherit Palatino and Times settings respectively, also the TEX Gyre fonts Pagella and Termes (formerly: qfonts).

```
5740 %% -- Palatino
5741 \DeclareMicrotypeAlias{pxr} {ppl}          % pxfonts
5742 \DeclareMicrotypeAlias{qpl} {ppl}          % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)
```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```
5743 \DeclareMicrotypeAlias{fp9x}{pplx}         % FPL Neu
5744 \DeclareMicrotypeAlias{fp9j}{pplj}         %     "
```

The `newpx` package, a replacement for `pxfonts`.

```
5745 \DeclareMicrotypeAlias{zpllf}{pplx}        % newpxtext
5746 \DeclareMicrotypeAlias{zplosf}{pplj}       %     "
5747 \DeclareMicrotypeAlias{zpltlf}{pplx}       %     "
5748 \DeclareMicrotypeAlias{zpltosf}{pplj}      %     "
```

The `domitian` package.

```
5749 \DeclareMicrotypeAlias{Domitian-TLF} {pplx}% domitian
5750 \DeclareMicrotypeAlias{Domitian-TOsF}{pplj}%     "
```

The OpenType versions:

```
5751 \DeclareMicrotypeAlias{Palatino Linotype}{Palatino}
5752 \DeclareMicrotypeAlias{Palatino LT Std}  {Palatino}
5753 \DeclareMicrotypeAlias{TeX Gyre Pagella} {Palatino}
5754 \DeclareMicrotypeAlias{Domitian}         {Palatino}
5755 \DeclareMicrotypeAlias{Asana Math}       {Palatino}
5756 %% -- Times New Roman
5757 \DeclareMicrotypeAlias{txr}{ptm}          % txfonts
```

The `newtx` package, a replacement for `txfonts`.

```
5758 \DeclareMicrotypeAlias{ntxlf}{ptmx}        % newtxtext
5759 \DeclareMicrotypeAlias{ntxosf}{ptmj}       %     "
5760 \DeclareMicrotypeAlias{ntxtlf}{ptmx}       %     "
5761 \DeclareMicrotypeAlias{ntxtosf}{ptmj}      %     "
```

The `tempora` package.

```
5762 \DeclareMicrotypeAlias{Tempora-TLF} {ptmx} % tempora
5763 \DeclareMicrotypeAlias{Tempora-TOsF}{ptmj} %     "
5764 \DeclareMicrotypeAlias{qtm}{ptm}          % TeX Gyre Termes (formerly: qfonts/QuasiTimes)
```

The `step` package.

```
5765 \DeclareMicrotypeAlias{STEP-TLF} {ptmx}    % step
5766 \DeclareMicrotypeAlias{STEP-TOsF}{ptmj}    %     "
```

The `stix` and `stix2` packages (the latter has departed a bit from being a Times clone, but still seems close enough).

```
5767 \DeclareMicrotypeAlias{stix} {ptm}         % stix
5768 \DeclareMicrotypeAlias{stix2}{ptm}         % stix2
```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (Times-NewRomanMT, TimesNRSevenMT), mtm (TimesSmallTextMT); pte (TimesEuropa); ptt (TimesTen); TimesEighteen; TimesModernEF.

MicroPress's Charter version (chmath).

```
5769 %% -- Charter
5770 \DeclareMicrotypeAlias{chr}{bch}          % CH Math
```

The XCharter package extends the Charter fonts.

```
5771 \DeclareMicrotypeAlias{XCharter-TLF} {bch} % XCharter
5772 \DeclareMicrotypeAlias{XCharter-TOsF}{bch} %    "
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
5773 \DeclareMicrotypeAlias{mdbch}{bch}        % mathdesign/Charter
5774 %% -- Garamond
5775 \DeclareMicrotypeAlias{mdugm}{ugm}        % mathdesign/URW Garamond
```

The garamondx package, an extension of URW Garamond, providing small caps and oldstyle figures.

```
5776 \DeclareMicrotypeAlias{zgmx}{ugm}         % garamondx
5777 \DeclareMicrotypeAlias{zgmj}{ugm}         %    "
5778 \DeclareMicrotypeAlias{zgmI}{ugm}         %    "
5779 \DeclareMicrotypeAlias{zgmq}{ugm}         %    "
```

Because a configuration file for Adobe Garamond wouldn't be permitted for TeX Live distribution, we use EB Garamond as the base font.

```
5780 \DeclareMicrotypeAlias{pad} {EBGaramond-LF}% Adobe Garamond
5781 \DeclareMicrotypeAlias{padx}{EBGaramond-TLF}%    "
5782 \DeclareMicrotypeAlias{padj}{EBGaramond-TOsF}%   "
5783 %% --
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
5784 \DeclareMicrotypeAlias{ulg}{blg}          % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

The eulervm package virtually extends the Euler fonts.

```
5785 \DeclareMicrotypeAlias{zeur}{eur}         % Euler VM
5786 \DeclareMicrotypeAlias{zeus}{eus}         %    "
```

Euro symbol fonts, to save some files.

```
5787 \DeclareMicrotypeAlias{zpeus}   {zpeu}    % Adobe Euro sans -> serif
5788 \DeclareMicrotypeAlias{eurosans}{zpeu}    % Adobe Euro sans -> serif
```

The Lato and Fontin fonts (and many, many more...) only contain a basic set of glyphs. We alias them here to the basic settings (see 3.1.5) to prevent lots of warning messages from the inheritance settings; they will still receive protrusion settings from the default (T1) configuration.

```
5789 \DeclareMicrotypeAlias{Lato}            {TU-basic}
5790 \DeclareMicrotypeAlias{Lato-Regular}   {TU-basic}
5791 \DeclareMicrotypeAlias{Fontin}         {TU-basic}
5792 \DeclareMicrotypeAlias{Fontin-Regular} {TU-basic}
5793 \DeclareMicrotypeAlias{Bergamo Std}    {TU-basic}
```

The fontawesome and fontawesome5 packages are aliased to empty settings (see 3.1.6 and 3.2.6).

```
5794 \DeclareMicrotypeAlias{FontAwesome}       {TU-empty} % fontawesome
5795 \DeclareMicrotypeAlias{fontawesomefree}   {TU-empty} % fontawesome5
5796 \DeclareMicrotypeAlias{fontawesomepro}    {TU-empty}
5797 \DeclareMicrotypeAlias{fontawesomebrands}{TU-empty}
5798
```

2.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
5799 %%% ----------------------------------------------------------------------
5800 %%% INTERACTION WITH THE `babel' PACKAGE
5801
5802 \DeclareMicrotypeBabelHook
5803    {english,UKenglish,british,USenglish,american}
5804    {kerning=, spacing=nonfrench}
5805
5806 \DeclareMicrotypeBabelHook
5807    {french,francais,acadian,canadien}
5808    {kerning=french, spacing=}
5809
5810 \DeclareMicrotypeBabelHook
5811    {turkish}
5812    {kerning=turkish, spacing=}
5813
```

2.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\   :   \textbackslash
{   :   \textbraceleft
}   :   \textbraceright
^   :   \textasciicircum
%   :   \%
#   :   \#
```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper LaTeX way, that is, when they have been assigned a slot in the font encoding with \DeclareTextSymbol or \DeclareTextComposite. Characters defined via \chardef are also possible.

Ligatures and \mathchardef'ed symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed, or else in combination with the 'inputenc' key.

With XƎTEX or LuaTEX, in contrast, it is advisable to use the proper Unicode characters, or the font-specific glyph names prefixed with '/' (cf. section 3).

2.5 Character inheritance

First the lists of inheriting characters. We only declare those characters that are the same on *both* sides, i.e., not Œ for O.

```
5814 ⟨/m-t⟩
5815 ⟨*m-t|ebg|zpeu|mvs⟩
5816 %%% ----------------------------------------------------------------------
5817 %%% CHARACTER INHERITANCE
5818
```

```
5819  ⟨/m-t|ebg|zpeu|mvs⟩
5820  ⟨*m-t⟩
```

2.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 ('fi' ligature), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
5821  \DeclareCharacterInheritance
5822    { encoding = OT1 }
5823    { f = {011}, % ff
5824      i = {\i},
5825      j = {\j},
5826      O = {\O},
5827      o = {\o}
5828    }
5829
```

2.5.2 T1

Candidates here: 028 ('fi'), 029 ('fl'), 030 ('ffi'), 031 ('ffl'), 156 ('IJ' ligature, since LaTeX 2005/12/01 accessible as \IJ), 188 ('ij', \ij), Æ, æ, Œ, œ.

```
5830  \DeclareCharacterInheritance
5831    { encoding = T1 }
5832    { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
5833      a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
5834      C = {\'C,\c C,\v C},
5835      c = {\'c,\c c,\v c},
5836      D = {\v D,\DH},
5837      d = {\v d,\dj},
5838      E = {\`E,\'E,\^E,\"E,\k E,\v E},
5839      e = {\`e,\'e,\^e,\"e,\k e,\v e},
5840      f = {027}, % ff
5841      G = {\u G},
5842      g = {\u g},
5843      I = {\`I,\'I,\^I,\"I,\.I},
5844      i = {\`i,\'i,\^i,\"i,\i},
5845      j = {\j},
5846      L = {\L,\'L,\v L},
5847      l = {\l,\'l,\v l},
5848      N = {\'N,\~N,\v N},
5849      n = {\'n,\~n,\v n},
5850      O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
5851      o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
5852      R = {\'R,\v R},
5853      r = {\'r,\v r},
5854      S = {\'S,\c S,\v S,\SS},
5855      s = {\'s,\c s,\v s},
5856      T = {\c T,\v T},
5857      t = {\c t,\v t},
5858      U = {\`U,\'U,\^U,\"U,\H U,\r U},
5859      u = {\`u,\'u,\^u,\"u,\H u,\r u},
5860      Y = {\'Y,\"Y},
5861      y = {\'y,\"y},
5862      Z = {\'Z,\.Z,\v Z},
5863      z = {\'z,\.z,\v z}
```

The 'soft hyphen' often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```
5864  %     - = {127},
5865    }
5866
```

2.5.3 LY1

More characters: 008 ('fl'), 012 ('fi'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
5867  \DeclareCharacterInheritance
5868    { encoding = LY1 }
5869    { A = {\`A,\'A,\^A,\~A,\"A,\r A},
5870      a = {\`a,\'a,\^a,\~a,\"a,\r a},
5871      C = {\c C},
5872      c = {\c c},
5873      D = {\DH},
5874      E = {\`E,\'E,\^E,\"E},
5875      e = {\`e,\'e,\^e,\"e},
5876      f = {011}, % ff
5877      I = {\`I,\'I,\^I,\"I},
5878      i = {\`i,\'i,\^i,\"i,\i},
5879      L = {\L},
5880      l = {\l},
5881      N = {\~N},
5882      n = {\~n},
5883      O = {\`O,\'O,\^O,\~O,\"O,\O},
5884      o = {\`o,\'o,\^o,\~o,\"o,\o},
5885      S = {\v S},
5886      s = {\v s},
5887      U = {\`U,\'U,\^U,\"U},
5888      u = {\`u,\'u,\^u,\"u},
5889      Y = {\'Y,\"Y},
5890      y = {\'y,\"y},
5891      Z = {\v Z},
5892      z = {\v z}
5893    }
5894
```

2.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 ('fk'), 012 ('fi'),
013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
5895  \DeclareCharacterInheritance
5896    { encoding = OT4 }
5897    { A = {\k A},
5898      a = {\k a},
5899      C = {\'C},
5900      c = {\'c},
5901      E = {\k E},
5902      e = {\k e},
5903      f = {011}, % ff
5904      i = {\i},
5905      j = {\j},
5906      L = {\L},
5907      l = {\l},
5908      N = {\'N},
5909      n = {\'n},
5910      O = {\O,\'O},
5911      o = {\o,\'o},
5912      S = {\'S},
5913      s = {\'s},
5914      Z = {\'Z,\.Z},
5915      z = {\'z,\.z},
5916      \textquotedblleft = "FF
5917    }
5918
```

2.5.5 QX

The Central European QX encoding.[6] Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'),
014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
5919  \DeclareCharacterInheritance
5920    { encoding = QX }
5921    { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
5922      a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
5923      C = {\'C,\c C},
5924      c = {\'c,\c c},
5925      D = {\DH},
5926      E = {\`E,\'E,\^E,\"E,\k E},
5927      e = {\`e,\'e,\^e,\"e,\k e},
5928      f = {011}, % ff
5929      I = {\`I,\'I,\^I,\"I,\k I},
5930      i = {\`i,\'i,\^i,\"i,\k i,\i},
5931      j = {\j},
5932      L = {\L},
5933      l = {\l},
5934      N = {\'N,\~N},
5935      n = {\'n,\~n},
5936      O = {\O,\`O,\'O,\^O,\~O,\"O},
5937      o = {\o,\`o,\'o,\^o,\~o,\"o},
```

The Romanian \textcommabelow accents are actually replacements for the \c
variants, which had previously (and erroneously[7]) been included in QX encoding.
They are still kept for backwards compatibility.

```
5938      S = {\'S,\c S,\textcommabelow S,\v S},
5939      s = {\'s,\c s,\textcommabelow s,\v s},
5940      T = {\c T,\textcommabelow T},
5941      t = {\c t,\textcommabelow t},
5942      U = {\`U,\'U,\^U,\"U,\k U},
5943      u = {\`u,\'u,\^u,\"u,\k u},
5944      Y = {\'Y,\"Y},
5945      y = {\'y,\"y},
5946      Z = {\'Z,\.Z,\v Z},
5947      z = {\'z,\.z,\v z},
5948      . = \textellipsis
5949    }
5950
```

2.5.6 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented
characters that there is no room for any ligatures.

```
5951  \DeclareCharacterInheritance
5952    { encoding = T5 }
5953    { A = {\`A,\'A,\~A,\h A,\d A,\^A,\u A,
5954          \`\Acircumflex,\'\Acircumflex,\~\Acircumflex,\h\Acircumflex,\d\Acircumflex,
5955          \`\Abreve,\'\Abreve,\~\Abreve,\h\Abreve,\d\Abreve},
5956      a = {\`a,\'a,\~a,\h a,\d a,\^a,\u a,
5957          \`\acircumflex,\'\acircumflex,\~\acircumflex,\h\acircumflex,\d\acircumflex,
5958          \`\abreve,\'\abreve,\~\abreve,\h\abreve,\d\abreve},
5959      D = {\DJ},
5960      d = {\dj},
5961      E = {\`E,\'E,\~E,\h E,\d E,\^E,
5962          \`\Ecircumflex,\'\Ecircumflex,\~\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
5963      e = {\`e,\'e,\~e,\h e,\d e,\^e,
5964          \`\ecircumflex,\'\ecircumflex,\~\ecircumflex,\h\ecircumflex,\d\ecircumflex},
```

---

6 Contributed by *Maciej Eder*.
7 Cf. http://tug.org/pipermail/tex-live/2008-August/017204.html

```
5965      I = {\`I,\'I,\~I,\h I,\d I},
5966      i = {\`i,\'i,\~i,\h i,\d i,\i},
5967      O = {\`O,\'O,\~O,\h O,\d O,\^O,\horn O,
5968          \`\Ocircumflex,\'\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
5969          \`\Ohorn,\'\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},
5970      o = {\`o,\'o,\~o,\h o,\d o,\^o,\horn o,
5971          \`\ocircumflex,\'\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
5972          \`\ohorn,\'\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
5973      U = {\`U,\'U,\~U,\h U,\d U,\horn U,
5974          \`\Uhorn,\'\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
5975      u = {\`u,\'u,\~u,\h u,\d u,\horn u,
5976          \`\uhorn,\'\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
5977      Y = {\`Y,\'Y,\~Y,\h Y,\d Y},
5978      y = {\`y,\'y,\~y,\h y,\d y}
5979    }
5980
```

2.5.7 EU1, EU2, TU

The EU1 (X꜍TEX), EU2 (LuaTEX), and, since `fontspec` version 2.5, TU encodings
are not well-defined in the sense that they don't contain a fixed number of glyphs,
all of which must be present. OpenType fonts may contain thousands of glyphs,
but we only define those that should be present in every font (basically T1). This
inheritance list should be overridden by font-specific ones.

```
5981 \DeclareCharacterInheritance
5982    { encoding = {TU,EU1,EU2} }
5983    { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
5984      a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
5985      C = {\'C,\c C,\v C},
5986      c = {\'c,\c c,\v c},
5987      D = {\v D,\DH},
5988      d = {\v d,\dj},
5989      E = {\`E,\'E,\^E,\"E,\k E,\v E},
5990      e = {\`e,\'e,\^e,\"e,\k e,\v e},
5991 %    f = {/f_f}, % sometimes /f_f, sometimes /ff
5992      G = {\u G},
5993      g = {\u g},
5994      I = {\`I,\'I,\^I,\"I,\.I},
5995      i = {\`i,\'i,\^i,\"i,\i},
5996 %    j = {\j},
5997      L = {\L,\'L,\v L},
5998      l = {\l,\'l,\v l},
5999      N = {\'N,\~N,\v N},
6000      n = {\'n,\~n,\v n},
6001      O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
6002      o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
6003      R = {\'R,\v R},
6004      r = {\'r,\v r},
6005      S = {\'S,\c S,\v S}, % \SS
6006      s = {\'s,\c s,\v s},
6007      T = {\c T,\v T},
6008      t = {\c t,\v t},
6009      U = {\`U,\'U,\^U,\"U,\H U,\r U},
6010      u = {\`u,\'u,\^u,\"u,\H u,\r u},
6011      Y = {\'Y,\"Y},
6012      y = {\'y,\"y},
6013      Z = {\'Z,\.Z,\v Z},
6014      z = {\'z,\.z,\v z}
6015    }
6016
6017 ⟨/m-t⟩
```

2.5.8 LGR

The Greek LGR encoding. EB Garamond contains some more glyphs.

```
6018 ⟨*m-t|ebg⟩
6019 \DeclareCharacterInheritance
6020   { encoding = LGR,
6021 ⟨ebg⟩    family = {EBGaramond-OsF,EBGaramond-TOsF,EBGaramond-LF,EBGaramond-TLF}
6022   }
6023   {
6024 ⟨m-t⟩    A = {012},
6025 ⟨ebg⟩    A = {009,012,253},
6026 ⟨ebg⟩ (1)E = {199},
6027 ⟨ebg⟩    H = {010},
6028 ⟨ebg⟩ (1)H = {159},
6029      I = {219},
6030 ⟨ebg⟩ (1)I = {155},
6031      O = J,
6032 ⟨ebg⟩ (1)O = {151},
6033      U = {013,223},
6034      W = {011},
6035      a = {014,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,
6036          144,145,146,148,149,150,248},
6037      e = {224,225,226,227,232,233,234,235},
6038      h = {152,153,154,156,157,158,160,161,162,163,164,165,166,167,168,169,170,
6039          171,172,173,174,175,249},
6040 ⟨m-t⟩    i = {200,201,202,203,208,209,210,211,216,217,218,240,241,242,243},
6041 ⟨ebg⟩    i = {008,200,201,202,203,208,209,210,211,216,217,218,240,241,242,243},
6042      o = {228,229,230,231,236,237,238,239},
6043      r = {251,252},
6044      u = {015,204,205,206,207,212,213,214,215,220,221,222,244,245,246,247},
6045      w = {176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,
6046          193,194,196,197,198,250},
6047 ⟨ebg⟩    \textstigma = \textvarstigma,
6048      . = {059}  % ano teleia
6049   }
6050
6051 ⟨/m-t|ebg⟩
```

2.5.9 Euro symbols

Make Euro symbols settings simpler.

```
6052 ⟨*zpeu⟩
6053 \DeclareCharacterInheritance
6054   { encoding = U,
6055     family  = {zpeu,zpeus,eurosans} }
6056   { E = 128 }
6057
6058 ⟨/zpeu⟩
6059 ⟨*mvs⟩
```

Since 2006/05/11 (that is, one week after I've added these settings, after the package had been dormant for six years!), marvosym's encoding is (correctly) U instead of OT1.

```
6060 \DeclareCharacterInheritance
6061   { encoding = {OT1,U},
6062     family  = mvs }
6063   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
6064
6065 ⟨/mvs⟩
```

2.6 Tracking

By default, we only disable the 'f*' ligatures, for those fonts that have any. Thus, ligatures and especially kerning for all other characters will be retained.

```
6066 ⟨*m-t⟩
6067 %%% ----------------------------------------------------------------------
6068 %%% TRACKING/LETTERSPACING
6069
6070 \SetTracking
6071  [ name        = default,
6072    no ligatures = {f} ]
6073  { encoding     = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
6074  { }
6075
```

2.7 Font expansion

These are Hàn Thế Thành's original expansion settings. They are used for all fonts (until somebody shows mercy and creates font-specific settings).

```
6076 %%% ----------------------------------------------------------------------
6077 %%% EXPANSION
6078
6079 \SetExpansion
6080  [ name      = default      ]
6081  { encoding = {OT1,OT4,QX,T1,LY1} }
6082  {
6083    A = 500,      a = 700,
6084  \AE = 500,    \ae = 700,
6085    B = 700,      b = 700,
6086    C = 700,      c = 700,
6087    D = 500,      d = 700,
6088    E = 700,      e = 700,
6089    F = 700,
6090    G = 500,      g = 700,
6091    H = 700,      h = 700,
6092    K = 700,      k = 700,
6093    M = 700,      m = 700,
6094    N = 700,      n = 700,
6095    O = 500,      o = 700,
6096  \OE = 500,    \oe = 700,
6097    P = 700,      p = 700,
6098    Q = 500,      q = 700,
6099    R = 700,
6100    S = 700,      s = 700,
6101    U = 700,      u = 700,
6102    W = 700,      w = 700,
6103    Z = 700,      z = 700,
6104    2 = 700,
6105    3 = 700,
6106    6 = 700,
6107    8 = 700,
6108    9 = 700
6109  }
6110
```

Settings for Cyrillic T2A encoding. [8]

```
6111 \SetExpansion
6112  [ name      = T2A ]
6113  { encoding = T2A }
6114  {
6115    A = 500,      a = 700,
```

---

[8] Contributed by *Karl Karlsson*.

```
6116      B = 700,      b = 700,
6117      C = 700,      c = 700,
6118      D = 500,      d = 700,
6119      E = 700,      e = 700,
6120      F = 700,
6121      G = 500,      g = 700,
6122      H = 700,      h = 700,
6123      K = 700,      k = 700,
6124      M = 700,      m = 700,
6125      N = 700,      n = 700,
6126      O = 500,      o = 700,
6127      P = 700,      p = 700,
6128      Q = 500,      q = 700,
6129      R = 700,
6130      S = 700,      s = 700,
6131      U = 700,      u = 700,
6132      W = 700,      w = 700,
6133      Z = 700,      z = 700,
6134      2 = 700,
6135      3 = 700,
6136      6 = 700,
6137      8 = 700,
6138      9 = 700,
6139      \CYRA = 500,     \cyra = 700,
6140      \CYRB = 700,     \cyrb = 700,
6141      \CYRV = 700,     \cyrv = 700,
6142      \CYRG = 700,     \cyrg = 700,
6143      \CYRD = 700,     \cyrd = 700,
6144      \CYRE = 700,     \cyre = 700,
6145      \CYRZH = 700,    \cyrzh = 700,
6146      \CYRZ = 700,     \cyrz = 700,
6147      \CYRI = 700,     \cyri = 700,
6148      \CYRISHRT = 700, \cyrishrt = 700,
6149      \CYRK = 700,     \cyrk = 700,
6150      \CYRL = 700,     \cyrl = 700,
6151      \CYRM = 700,     \cyrm = 700,
6152      \CYRN = 700,     \cyrn = 700,
6153      \CYRO = 500,     \cyro = 700,
6154      \CYRP = 700,     \cyrp = 700,
6155      \CYRR = 700,     \cyrr = 700,
6156      \CYRS = 700,     \cyrs = 700,
6157      \CYRT = 700,     \cyrt = 700,
6158      \CYRU = 700,     \cyru = 700,
6159      \CYRF = 700,     \cyrf = 700,
6160      \CYRH = 700,     \cyrh = 700,
6161      \CYRC = 700,     \cyrc = 700,
6162      \CYRCH = 700,    \cyrch = 700,
6163      \CYRSH = 700,    \cyrsh = 700,
6164      \CYRSHCH = 700,  \cyrshch = 700,
6165      \CYRHRDSN = 700, \cyrhrdsn = 700,
6166      \CYRERY = 700,   \cyrery = 700,
6167      \CYRSFTSN = 700, \cyrsftsn = 700,
6168      \CYREREV = 700,  \cyrerev = 700,
6169      \CYRYU = 700,    \cyryu = 700,
6170      \CYRYA = 700,    \cyrya = 700
6171    }
6172
```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```
6173 \SetExpansion
6174    [ name     = T5 ]
6175    { encoding = T5 }
6176    {
6177      A = 500,      a = 700,
6178      B = 700,      b = 700,
```

```
6179        C = 700,      c = 700,
6180        D = 500,      d = 700,
6181        E = 700,      e = 700,
6182        F = 700,
6183        G = 500,      g = 700,
6184        H = 700,      h = 700,
6185        K = 700,      k = 700,
6186        M = 700,      m = 700,
6187        N = 700,      n = 700,
6188        O = 500,      o = 700,
6189        P = 700,      p = 700,
6190        Q = 500,      q = 700,
6191        R = 700,
6192        S = 700,      s = 700,
6193        U = 700,      u = 700,
6194        W = 700,      w = 700,
6195        Z = 700,      z = 700,
6196        2 = 700,
6197        3 = 700,
6198        6 = 700,
6199        8 = 700,
6200        9 = 700
6201      }
6202
6203  ⟨/m-t⟩
```

2.8 Character protrusion

```
6204  %%% ------------------------------------------------------------------------
6205  %%% PROTRUSION
6206
```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to `microtype` notation).

```
\SetProtrusion
  [ name      = thanh ]
  { encoding = OT1 }
  {
    A = {50,50},
    F = {  ,50},
    J = {50,  },
    K = {  ,50},
    L = {  ,50},
    T = {50,50},
    V = {50,50},
    W = {50,50},
    X = {50,50},
    Y = {50,50},
    k = {  ,50},
    r = {  ,50},
    t = {  ,50},
    v = {50,50},
    w = {50,50},
    x = {50,50},
    y = {50,50},
    . = {  ,700},    {,}= { ,700},
    : = {  ,500},      ; = { ,500},
    ! = {  ,200},      ? = { ,200},
    ( = {50,  },      ) = {   ,50},
    - = {  ,700},
    \textendash       = {  ,300},      \textemdash         = {  ,200},
    \textquoteleft    = {700, },       \textquoteright     = {  ,700},
    \textquotedblleft = {500, },       \textquotedblright  = {  ,500}
  }
```

2.8.1 Normal

The default settings always use the most moderate value.

```
6207 ⟨*cfg-t⟩
6208 \SetProtrusion
6209 ⟨m-t⟩   [ name     = default ]
```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```
6210 ⟨bch⟩   [ name     = bch-default ]
```

- Bitstream Letter Gothic (blg)

```
6211 ⟨blg⟩   [ name     = blg-default ]
```

- Computer Modern Roman (cmr)

```
6212 ⟨cmr⟩   [ name     = cmr-default ]
```

- EB Garamond

```
6213 ⟨ebg⟩   [ name     = EBGaramond-default ]
```

- Minion [9] (pmnx, pmnj)

```
6214 ⟨pmn⟩   [ name     = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
6215 ⟨ppl⟩   [ name     = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
6216 ⟨ptm⟩   [ name     = ptm-default ]
```

- URW Garamond (ugm)

```
6217 ⟨ugm⟩   [ name     = ugm-default ]
6218 ⟨m-t|cmr|pmn|ebg⟩   { }
6219 ⟨bch|blg|ugm⟩   { encoding = OT1,
6220 ⟨ppl|ptm⟩   { encoding = {OT1,OT4},
6221 ⟨bch⟩     family   = bch }
6222 ⟨blg⟩     family   = blg }
6223 ⟨ppl⟩     family   = {ppl,pplx,pplj} }
6224 ⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
6225 ⟨ugm⟩     family   = ugm }
6226     {
6227 ⟨m-t|bch|blg|cmr|ebg|pmn|ppl|ptm⟩      A = {50,50},
6228 ⟨ugm⟩     A = {50,100},
6229 ⟨ebg|ptm⟩   \AE = {50,  },
6230 ⟨ugm⟩   \AE = {150,50},
6231 ⟨ugm⟩     B = {  ,50},
6232 ⟨bch|ebg|pmn|ugm⟩     C = {50,  },
6233 ⟨bch|ebg|pmn⟩     D = {  ,50},
6234 ⟨ugm⟩     D = {  ,70},
6235 ⟨ugm⟩     E = {  ,50},
6236 ⟨m-t|bch|cmr|ebg|pmn|ptm⟩     F = {  ,50},
6237 ⟨ugm⟩     F = {  ,70},
6238 ⟨bch|ebg|pmn⟩     G = {50,  },
6239 ⟨ugm⟩     G = {50,50},
6240 ⟨blg⟩     I = {150,150},
6241 ⟨m-t|cmr|ebg|pmn|ppl|ptm|ugm⟩     J = {50,  },
6242 ⟨bch|blg⟩     J = {100,  },
```

---

9   Contributed by *Harald Harders* and *Karl Karlsson*.

```
6243 ⟨!blg⟩      K = {  ,50},
6244 ⟨blg⟩       K = {50,  },
6245 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩     L = {  ,50},
6246 ⟨blg⟩       L = {  ,150},
6247 ⟨ptm⟩       L = {  ,80},
6248 ⟨ugm⟩       L = {  ,120},
6249 ⟨bch|ebg|pmn|ugm⟩     O = {50,50},
6250 ⟨ebg⟩      \OE = {50,  },
6251 ⟨ugm⟩      \OE = {50,50},
6252 ⟨blg⟩       P = {  ,100},
6253 ⟨ugm⟩       P = {  ,50},
6254 ⟨bch|ebg|pmn⟩     Q = {50,70},
6255 ⟨ugm⟩       Q = {50,50},
6256 ⟨bch⟩       R = {  ,50},
6257 ⟨ugm|ebg⟩     R = {  ,70},
6258 ⟨m-t|bch|cmr|pmn|ppl|ptm⟩     T = {50,50},
6259 ⟨blg⟩       T = {100,100},
6260 ⟨ebg|ugm⟩     T = {70,70},
6261 ⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩     V = {50,50},
6262 ⟨blg|ugm⟩     V = {70,70},
6263 ⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩     W = {50,50},
6264 ⟨ugm⟩       W = {70,70},
6265 ⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩     X = {50,50},
6266 ⟨ugm⟩       X = {50,70},
6267 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩     Y = {50,50},
6268 ⟨blg|ptm|ugm⟩     Y = {80,80},
6269 ⟨ugm⟩       Z = {50,50},
6270 ⟨blg⟩       f = {150,100},
6271 ⟨blg⟩       i = {150,150},
6272 ⟨blg⟩       j = {100,100},
6273 ⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩     k = {  ,50},
6274 ⟨ugm⟩       k = {  ,70},
6275 ⟨blg⟩       l = {150,150},
6276 ⟨pmn⟩       l = {  ,-50},
6277 ⟨ppl⟩       p = {50,50},
6278 ⟨ebg|ugm⟩     p = {  ,50},
6279 ⟨ebg|ppl⟩     q = {50,  },
6280 ⟨!blg⟩      r = {  ,50},
6281 ⟨blg⟩       r = {100, 80},
6282 ⟨cmr|ebg|pmn⟩     t = {  ,70},
6283 ⟨bch⟩       t = {  ,50},
6284 ⟨blg⟩       t = {150, 80},
6285 ⟨ugm⟩       t = {  ,100},
6286 ⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩     v = {50,50},
6287 ⟨blg⟩       v = {100,100},
6288 ⟨ugm⟩       v = {50,70},
6289 ⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩     w = {50,50},
6290 ⟨ugm⟩       w = {50,70},
6291 ⟨!blg⟩      x = {50,50},
6292 ⟨blg⟩       x = {100,100},
6293 ⟨m-t|bch|ebg|pmn⟩     y = {  ,50},
6294 ⟨blg⟩       y = { 50,100},
6295 ⟨cmr|ppl|ptm⟩     y = {50,70},
6296 ⟨ugm⟩       y = {  ,70},

6297 ⟨cmr⟩       0 = {  ,50},
6298 ⟨m-t⟩       1 = {50,50},
6299 ⟨bch|blg|ptm|ugm⟩     1 = {150,150},
6300 ⟨cmr⟩       1 = {100,200},
6301 ⟨pmn⟩       1 = {  ,50},
6302 ⟨ppl⟩       1 = {100,100},
6303 ⟨bch|cmr|ugm⟩     2 = {50,50},
6304 ⟨blg⟩       2 = {  ,100},
6305 ⟨bch|pmn⟩     3 = {50,  },
6306 ⟨cmr|ugm⟩     3 = {50,50},
6307 ⟨blg⟩       3 = {100,  },
```

```
6308 ⟨m-t⟩      4 = {50,50},
6309 ⟨bch⟩      4 = {100,50},
6310 ⟨blg⟩      4 = {100,  },
6311 ⟨cmr|ugm⟩     4 = {70,70},
6312 ⟨pmn⟩      4 = {50,  },
6313 ⟨ptm⟩      4 = {70,  },
6314 ⟨cmr⟩      5 = {  ,50},
6315 ⟨bch⟩      6 = {50,  },
6316 ⟨cmr⟩      6 = {  ,50},
6317 ⟨m-t⟩      7 = {50,50},
6318 ⟨bch|pmn|ugm⟩     7 = {50,80},
6319 ⟨blg⟩      7 = {100,100},
6320 ⟨cmr|ptm⟩     7 = {50,100},
6321 ⟨ppl⟩      7 = {  ,50},
6322 ⟨cmr⟩      8 = {  ,50},
6323 ⟨bch⟩      9 = {50,50},
6324 ⟨cmr⟩      9 = {  ,50},
6325 ⟨m-t|cmr|pmn|ppl|ptm|ugm⟩       . = {  ,700},
6326 ⟨bch|ebg⟩       . = {  ,600},
6327 ⟨blg⟩      . = {400,500},
6328 ⟨!blg⟩      {,}= {  ,500},
6329 ⟨blg⟩      {,}= {300,400},
6330 ⟨m-t|cmr|pmn|ppl|ptm|ugm⟩      : = {  ,500},
6331 ⟨bch|ebg⟩      : = {  ,400},
6332 ⟨blg⟩      : = {300,400},
6333 ⟨m-t|bch|ebg|pmn|ptm⟩      ; = {  ,300},
6334 ⟨blg⟩      ; = {200,300},
6335 ⟨cmr|ppl⟩      ; = {  ,500},
6336 ⟨ugm⟩      ; = {  ,400},
6337 ⟨!blg⟩      ! = {  ,100},
6338 ⟨blg⟩      ! = {200,200},
6339 ⟨m-t|ebg|pmn|ptm⟩      ? = {  ,100},
6340 ⟨bch|cmr|ppl|ugm⟩      ? = {  ,200},
6341 ⟨blg⟩      ? = {150,150},
6342 ⟨pmn⟩      " = {300,300},
6343 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩      @ = {50,50},
6344 ⟨ptm⟩      @ = {100,100},
6345 ⟨m-t|bch|blg|cmr|ebg|pmn|ppl|ptm⟩      ~ = {200,250},
6346 ⟨ugm⟩      ~ = {300,350},
6347 ⟨ebg|ppl|ptm⟩      & = {50,100},
6348 ⟨ugm⟩      & = {  ,100},
6349 ⟨m-t|cmr|ebg|pmn⟩      \% = {50,50},
6350 ⟨bch⟩      \% = {  ,50},
6351 ⟨ppl|ptm⟩      \% = {100,100},
6352 ⟨ugm⟩      \% = {50,100},
6353 ⟨blg⟩      \# = {100,100},
6354 ⟨m-t|ppl|ptm|ugm⟩      * = {200,200},
6355 ⟨bch|pmn⟩      * = {200,300},
6356 ⟨blg⟩      * = {150,200},
6357 ⟨cmr|ebg⟩      * = {300,300},
6358 ⟨m-t|cmr|ebg|ppl|ptm⟩      + = {250,250},
6359 ⟨bch⟩      + = {150,250},
6360 ⟨blg|pmn⟩      + = {150,200},
6361 ⟨ugm⟩      + = {250,300},
6362 ⟨blg|ugm⟩      {=}= {200,200},
6363 ⟨m-t|ebg|pmn|ptm⟩      ( = {100,  },    ) = {   ,200},
6364 ⟨bch|ugm⟩      ( = {200,  },    ) = {   ,200},
6365 ⟨cmr|blg⟩      ( = {300,  },    ) = {   ,300},
6366 ⟨ppl⟩      ( = {100,  },    ) = {   ,300},
6367 ⟨bch|pmn⟩      [ = {100,  },    ] = {   ,100},
6368 ⟨blg⟩      [ = {300,100},    ] = {   ,300},

6369 ⟨m-t|ebg|pmn|ptm⟩      / = {100,200},
6370 ⟨bch⟩      / = {  ,200},
6371 ⟨blg⟩      / = {300,300},
6372 ⟨cmr|ppl⟩      / = {200,300},
```

```
6373 ⟨ugm⟩      / = {100,300},
6374 ⟨m-t|ptm⟩     - = {500,500},
6375 ⟨bch|cmr|ppl⟩    - = {400,500},
6376 ⟨blg⟩      - = {300,400},
6377 ⟨ebg⟩      - = {300,500},
6378 ⟨pmn⟩      - = {200,400},
6379 ⟨ugm⟩      - = {500,600},
6380 ⟨blg⟩      < = {200,100},    > = {100,200},
6381 ⟨blg⟩      _ = {150,250},
6382 ⟨blg⟩      | = {250,250},
6383 ⟨m-t|pmn⟩    \textendash       = {200,200},  \textemdash       = {150,150},
6384 ⟨bch⟩      \textendash     = {200,300},  \textemdash       = {150,250},
6385 ⟨cmr⟩      \textendash     = {400,300},  \textemdash       = {300,200},
6386 ⟨ebg|ppl|ptm⟩     \textendash      = {300,300},  \textemdash      = {200,200},
6387 ⟨ugm⟩      \textendash     = {250,300},  \textemdash       = {250,250},
```

Why settings for left *and* right quotes? Because in some languages they might be used like that (see the `csquotes` package for examples).

```
6388 ⟨m-t|bch|pmn⟩    \textquoteleft    = {300,400},  \textquoteright    = {300,400},
6389 ⟨blg⟩      \textquoteleft    = {400,600},  \textquoteright    = {400,600},
6390 ⟨cmr⟩      \textquoteleft    = {500,700},  \textquoteright    = {500,600},
6391 ⟨ebg⟩      \textquoteleft    = {300,500},  \textquoteright    = {400,400},
6392 ⟨ppl⟩      \textquoteleft    = {500,700},  \textquoteright    = {500,700},
6393 ⟨ptm⟩      \textquoteleft    = {500,500},  \textquoteright    = {300,500},
6394 ⟨ugm⟩      \textquoteleft    = {300,600},  \textquoteright    = {300,600},
6395 ⟨m-t|ebg|bch|pmn⟩     \textquotedblleft = {300,300},  \textquotedblright = {300,300}
6396 ⟨blg⟩      \textquotedblright = {300,400}
6397 ⟨cmr⟩      \textquotedblleft = {500,300},  \textquotedblright = {200,600}
6398 ⟨ppl|ptm⟩     \textquotedblleft = {300,400},  \textquotedblright = {300,400}
6399 ⟨ugm⟩      \textquotedblleft = {400,400},  \textquotedblright = {400,400}
6400    }
6401
```

Greek uppercase letters are in OT1 encoding only.

```
6402 ⟨*m-t|cmr|ebg|pmn⟩
6403 \SetProtrusion
6404 ⟨m-t⟩   [ name    = OT1-default,
6405 ⟨cmr⟩   [ name    = cmr-OT1,
6406 ⟨ebg⟩   [ name    = EBGaramond-OT1,
6407 ⟨pmn⟩   [ name    = pmnj-OT1,
6408 ⟨m-t⟩    load    = default ]
6409 ⟨cmr⟩    load    = cmr-default ]
6410 ⟨ebg⟩    load    = EBGaramond-default ]
6411 ⟨pmn⟩    load    = pmnj-default ]
6412 ⟨m-t⟩   { encoding = OT1 }
6413 ⟨cmr⟩   { encoding = {OT1,OT4},
6414 ⟨pmn⟩   { encoding = OT1,
6415 ⟨cmr⟩    family   = cmr  }
6416 ⟨pmn⟩    family   = pmnj }
6417 ⟨ebg⟩   { }
6418    {
6419 ⟨m-t|cmr⟩    \AE = {50,   },
6420 ⟨pmn⟩    \OE = {50,   }
6421 ⟨*cmr|ebg⟩
6422     "00 = {   ,150}, % \Gamma
6423     "01 = {100,100}, % \Delta
6424     "02 = { 50, 50}, % \Theta
6425     "03 = {100,100}, % \Lambda
6426 ⟨ebg⟩    "04 = { 50, 50}, % \Xi
6427 ⟨cmr⟩    "06 = { 50, 50}, % \Sigma
6428     "07 = {100,100}, % \Upsilon
6429     "08 = { 50, 50}, % \Phi
6430     "09 = { 50, 50}, % \Psi
6431 ⟨ebg⟩    "0A = { 50, 50}, % \Omega
6432 ⟨ebg⟩    138 = {   , 50}, % \L
```

Remaining slots can be found in the source file.

```
6433 ⟨/cmr|ebg⟩
6434    }
6435
```

Settings for figure variants.

```
6436 ⟨*ebg⟩
6437 \SetProtrusion
6438    [ name     = EBGaramond-OT1-LF,
6439      load     = EBGaramond-OT1 ]
6440    { encoding = OT1,
6441      family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
6442    {
6443      1 = {50,50},
6444      2 = {50,50},
6445      4 = {50,50},
6446      7 = {50,50},
6447    }
6448
6449 \SetProtrusion
6450    [ name     = EBGaramond-OT1-TOsF,
6451      load     = EBGaramond-OT1 ]
6452    { encoding = OT1,
6453      family   = {EBGaramond-TOsF} }
6454    {
6455      1 = {150,150},
6456      2 = {50,50},
6457      3 = {50,50},
6458      4 = {50,50},
6459      5 = {50,50},
6460      6 = {50,50},
6461      7 = {50,80},
6462      8 = {50,50},
6463      9 = {50,50},
6464    }
6465
6466 ⟨/ebg⟩
6467 ⟨/m-t|cmr|ebg|pmn⟩
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For X$_{\mathrm{H}}$TEX (EU1) and LuaTEX (EU2) we simply use the T1 list as default (for now).

```
6468 \SetProtrusion
6469 ⟨m-t⟩   [ name     = T1-default,
6470 ⟨bch⟩   [ name     = bch-T1,
6471 ⟨blg⟩   [ name     = blg-T1,
6472 ⟨cmr⟩   [ name     = cmr-T1,
6473 ⟨ebg⟩   [ name     = EBGaramond-T1,
6474 ⟨pmn⟩   [ name     = pmnj-T1,
6475 ⟨ppl⟩   [ name     = ppl-T1,
6476 ⟨ptm⟩   [ name     = ptm-T1,
6477 ⟨ugm⟩   [ name     = ugm-T1,
6478 ⟨m-t⟩     load     = default    ]
6479 ⟨bch⟩     load     = bch-default ]
6480 ⟨blg⟩     load     = blg-default ]
6481 ⟨cmr⟩     load     = cmr-default ]
6482 ⟨ebg⟩     load     = EBGaramond-default ]
6483 ⟨pmn⟩     load     = pmnj-default ]
6484 ⟨ppl⟩     load     = ppl-default ]
6485 ⟨ptm⟩     load     = ptm-default ]
6486 ⟨ugm⟩     load     = ugm-default ]
6487 ⟨m-t⟩   { encoding = {T1,LY1,EU1,EU2,TU} }
6488 ⟨bch|cmr|pmn|ppl⟩   { encoding = {T1,LY1},
6489 ⟨blg|ptm|ugm⟩   { encoding = {T1},
```

```
6490 ⟨ebg⟩    { encoding = {LY1},
6491 ⟨bch⟩      family   = bch }
6492 ⟨blg⟩      family   = blg }
6493 ⟨cmr⟩      family   = cmr }
6494 ⟨ebg⟩      family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF} }
6495 ⟨pmn⟩      family   = pmnj }
6496 ⟨ppl⟩      family   = {ppl,pplx,pplj} }
6497 ⟨ptm⟩      family   = {ptm,ptmx,ptmj} }
6498 ⟨ugm⟩      family   = ugm }
6499      {
6500 ⟨m-t|cmr⟩     \AE = {50,  },
6501 ⟨bch|pmn⟩     \OE = {50,  },
6502 ⟨pmn⟩     \TH = {  ,50},
6503 ⟨blg⟩     \v L = {   ,250},
6504 ⟨blg⟩     \v d = {   ,250},
6505 ⟨blg⟩     \v l = {   ,250},
6506 ⟨blg⟩     \v t = {   ,250},
6507 ⟨blg⟩     127 = {300,400},
6508 ⟨blg⟩     156 = {100,  }, % IJ
6509 ⟨blg⟩     188 = { 80, 80}, % ij
6510 ⟨m-t|bch|ebg|pmn|ppl|ptm⟩        _ = {100,100},
6511 ⟨cmr⟩         _ = {200,200},
6512 ⟨ugm⟩         _ = {100,200},
6513 ⟨m-t|ebg|pmn|ptm⟩    \textbackslash   = {100,200},
6514 ⟨bch⟩     \textbackslash   = {150,200},
6515 ⟨blg⟩     \textbackslash   = {250,300},
6516 ⟨cmr|ppl⟩     \textbackslash   = {200,300},
6517 ⟨ugm⟩     \textbackslash   = {100,300},
6518 ⟨ugm⟩     \textbar         = {200,200},
6519 ⟨blg⟩     \textendash      = {300,300},   \textemdash      = {150,150},
6520 ⟨blg⟩     \textquotedbl    = {300,400},   \textquotedblleft = {300,400},
6521 ⟨cmr⟩     \textquotedbl    = {300,300},   \textquotedblleft = {200,600},
```

The EC fonts do something weird: they insert an implicit kern between quote and boundary character. Therefore, we must override the settings from OT1.

```
6522 ⟨m-t|cmr|ebg|ppl|ptm|ugm⟩    \quotesinglbase  = {400,400},   \quotedblbase     = {400,400},
6523 ⟨blg⟩     \quotesinglbase  = {400,400},   \quotedblbase     = {300,400},
6524 ⟨bch|pmn⟩     \quotesinglbase  = {400,400},   \quotedblbase     = {300,300},
6525 ⟨m-t|bch|pmn⟩     \guilsinglleft   = {400,300},   \guilsinglright   = {300,400},
6526 ⟨blg⟩     \guilsinglleft   = {300,500},   \guilsinglright   = {300,500},
6527 ⟨cmr|ebg|ppl|ptm⟩     \guilsinglleft   = {400,400},   \guilsinglright   = {300,500},
6528 ⟨ugm⟩     \guilsinglleft   = {400,400},   \guilsinglright   = {300,600},
6529 ⟨m-t⟩     \guillemotleft   = {200,200},   \guillemotright   = {200,200},
6530 ⟨cmr⟩     \guillemotleft   = {300,200},   \guillemotright   = {100,400},
6531 ⟨bch|pmn⟩     \guillemotleft   = {200,200},   \guillemotright   = {150,300},
6532 ⟨blg|ppl|ptm⟩     \guillemotleft   = {300,300},   \guillemotright   = {200,400},
6533 ⟨ebg⟩     \guillemotleft   = {300,300},   \guillemotright   = {200,300},
6534 ⟨ugm⟩     \guillemotleft   = {300,400},   \guillemotright   = {300,400},
6535 ⟨m-t|bch|cmr|ebg|pmn|ppl|ugm⟩    \textexclamdown = {100,  },   \textquestiondown = {100,  },
6536 ⟨blg⟩     \textexclamdown = {200,  },   \textquestiondown = {100,  },
6537 ⟨ptm⟩     \textexclamdown = {200,  },   \textquestiondown = {200,  },
6538 ⟨m-t|cmr|ebg|ppl|ptm|ugm⟩    \textbraceleft   = {400,200},   \textbraceright   = {200,400},
6539 ⟨bch|blg|pmn⟩     \textbraceleft   = {200,  },   \textbraceright   = {  ,300},
6540 ⟨m-t|bch|cmr|ebg|ppl|ptm|ugm⟩    \textless        = {200,100},   \textgreater      = {100,200}
6541 ⟨pmn⟩     \textless        = {100,  },   \textgreater      = {  ,100},
6542 ⟨pmn⟩     \textvisiblespace = {100,100} % not in LY1
6543      }
6544
```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```
6545 ⟨*cmr⟩
6546 \SetProtrusion
6547    [ name      = lmr-T1,
```

```
6548       load    = cmr-T1    ]
6549     { encoding = {T1,LY1},
6550       family  = lmr       }
6551     {
6552       \textquotedblleft  = {300,400},  \textquotedblright = {300,400}
6553     }
6554
6555 ⟨/cmr⟩
6556 ⟨*ebg⟩
6557 \SetProtrusion
6558     [ name    = EBGaramond-T1-LF,
6559       load    = EBGaramond-T1 ]
6560     { encoding = T1,
6561       family  = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
6562     {
6563       1 = {50,50},
6564       2 = {50,50},
6565       4 = {50,50},
6566       7 = {50,50},
6567     }
6568
6569 \SetProtrusion
6570     [ name    = EBGaramond-T1-TOsF,
6571       load    = EBGaramond-T1 ]
6572     { encoding = T1,
6573       family  = {EBGaramond-TOsF} }
6574     {
6575       1 = {150,150},
6576       2 = {50,50},
6577       3 = {50,50},
6578       4 = {50,50},
6579       5 = {50,50},
6580       6 = {50,50},
6581       7 = {50,80},
6582       8 = {50,50},
6583       9 = {50,50},
6584     }
6585
6586 ⟨/ebg⟩
```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion). [10]

```
6587 ⟨*m-t|cmr|pmn⟩
6588 \SetProtrusion
6589 ⟨m-t⟩  [ name    = T2A-default,
6590 ⟨cmr⟩  [ name    = cmr-T2A,
6591 ⟨pmn⟩  [ name    = pmnj-T2A,
6592 ⟨m-t⟩    load    = default     ]
6593 ⟨cmr⟩    load    = cmr-default ]
6594 ⟨pmn⟩    load    = pmnj-default ]
6595    { encoding = T2A,
6596 ⟨m-t⟩   }
6597 ⟨cmr⟩     family  = cmr }
6598 ⟨pmn⟩     family  = pmnj }
6599    {
6600      \CYRA = {50,50},
6601      \CYRG = {  ,50},
6602      \CYRK = {  ,50},
6603      \CYRT = {50,50},
6604      \CYRH = {50,50},
6605      \CYRU = {50,50},
6606 ⟨pmn⟩     \CYRS = {50,  },
6607 ⟨pmn⟩     \CYRO = {50,50},
6608      \cyrk = {  ,50},
6609      \cyrg = {  ,50},
```

---

10 Contributed by *Karl Karlsson*.

```
6610        \cyrh = {50,50},
6611 ⟨m-t|pmn⟩      \cyru = {50,50},
6612 ⟨cmr⟩    \cyru = {50,70},
6613 ⟨m-t⟩          _  = {100,100},
6614 ⟨cmr⟩          _  = {200,200},
6615 ⟨m-t⟩     \textbackslash     = {100,200},   \quotedblbase       = {400,400},
6616 ⟨cmr⟩     \textbackslash     = {200,300},   \quotedblbase       = {400,400},
6617 ⟨pmn⟩     \textbackslash     = {100,200},   \quotedblbase       = {300,300},
6618 ⟨cmr⟩     \textquotedbl      = {300,300},   \textquotedblleft   = {200,600},
6619 ⟨m-t⟩     \guillemotleft     = {200,200},   \guillemotright     = {200,200},
6620 ⟨cmr⟩     \guillemotleft     = {300,200},   \guillemotright     = {100,400},
6621 ⟨pmn⟩     \guillemotleft     = {200,200},   \guillemotright     = {150,300},
6622 ⟨m-t|cmr⟩     \textbraceleft  = {400,200},   \textbraceright     = {200,400},
6623 ⟨pmn⟩     \textbraceleft     = {200, },     \textbraceright     = {  ,300},
6624 ⟨m-t|cmr⟩     \textless        = {200,100},   \textgreater        = {100,200}
6625 ⟨pmn⟩     \textless          = {100, },     \textgreater        = {  ,100}
6626     }
6627
6628 ⟨/m-t|cmr|pmn⟩
```

Settings for the QX encoding (generic and Times). [11] It also includes some glyphs otherwise in TS1.

```
6629 ⟨*m-t|ptm⟩
6630 \SetProtrusion
6631 ⟨m-t⟩   [ name     = QX-default,
6632 ⟨ptm⟩   [ name     = ptm-QX,
6633 ⟨m-t⟩     load     = default ]
6634 ⟨ptm⟩     load     = ptm-default ]
6635 ⟨m-t⟩   { encoding = QX }
6636 ⟨ptm⟩   { encoding = QX,
6637 ⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
6638    {
6639      \AE = {50,  },
6640 ⟨ptm⟩      *  = {200,200},
6641      {=} = {100,100},
6642      \textunderscore   = {100,100},
6643      \textbackslash    = {100,200},
6644      \quotedblbase     = {400,400},
6645 ⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
6646 ⟨ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
6647      \textexclamdown   = {100,  },   \textquestiondown = {100,  },
6648 ⟨m-t⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
6649 ⟨ptm⟩     \textbraceleft    = {200,200},   \textbraceright    = {200,300},
6650      \textless         = {200,100},   \textgreater       = {100,200},
6651      \textminus        = {200,200},   \textdegree        = {300,300},
6652 ⟨m-t⟩     \copyright        = {100,100},   \textregistered    = {100,100}
6653 ⟨ptm⟩     \copyright        = {100,150},   \textregistered    = {100,150},
6654 ⟨ptm⟩     \textxgeq         = {  ,100},   \textxleq          = {100,  },
6655 ⟨ptm⟩     \textalpha        = {  , 50},   \textDelta         = { 70, 70},
6656 ⟨ptm⟩     \textpi           = { 50, 80},   \textSigma         = {  , 70},
6657 ⟨ptm⟩     \textmu           = {  , 80},   \texteuro          = { 50, 50},
6658 ⟨ptm⟩     \textellipsis     = {150,200},   \textasciitilde    = { 80, 80},
6659 ⟨ptm⟩     \textapprox       = { 50, 50},   \textinfty         = {100,100},
6660 ⟨ptm⟩     \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
6661 ⟨ptm⟩     \textdiv          = { 50,150},   \textsection       = { 80, 80},
6662 ⟨ptm⟩     \texttimes        = {100,150},   \textpm            = { 50, 80},
6663 ⟨ptm⟩     \textbullet       = {150,150},   \textperiodcentered = {300,300},
6664 ⟨ptm⟩     \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
6665 ⟨ptm⟩     \textperthousand  = {  ,50}
6666    }
6667
6668 ⟨/m-t|ptm⟩
```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented

---

11   Contributed by *Maciej Eder*.

characters are already taken care of by the inheritance list.

```
6669 ⟨*cmr|bch⟩
6670 \SetProtrusion
6671 ⟨cmr⟩   [ name      = cmr-T5,
6672 ⟨cmr⟩      load      = cmr-default ]
6673 ⟨bch⟩   [ name      = bch-T5,
6674 ⟨bch⟩      load      = bch-default ]
6675    { encoding = T5,
6676 ⟨cmr⟩      family   = cmr }
6677 ⟨bch⟩      family   = bch }
6678      {
6679 ⟨bch⟩      _ = {100,100},
6680 ⟨bch⟩      \textbackslash    = {150,200},
6681 ⟨cmr⟩      \textbackslash    = {200,300},
6682 ⟨cmr⟩      \textquotedblleft = {200,600},
6683 ⟨cmr⟩      \textquotedbl     = {300,300},
6684 ⟨bch⟩      \quotesinglbase   = {400,400},   \quotedblbase    = {300,300},
6685 ⟨cmr⟩      \quotesinglbase   = {400,400},   \quotedblbase    = {400,400},
6686 ⟨bch⟩      \guilsinglleft    = {400,300},   \guilsinglright  = {300,400},
6687 ⟨cmr⟩      \guilsinglleft    = {400,400},   \guilsinglright  = {300,500},
6688 ⟨bch⟩      \guillemotleft    = {200,200},   \guillemotright  = {150,300},
6689 ⟨cmr⟩      \guillemotleft    = {300,200},   \guillemotright  = {100,400},
6690 ⟨bch⟩      \textbraceleft    = {200,   },   \textbraceright  = {   ,300},
6691 ⟨cmr⟩      \textbraceleft    = {400,200},   \textbraceright  = {200,400},
6692      \textless        = {200,100},   \textgreater        = {100,200}
6693      }
6694
6695 ⟨/cmr|bch⟩
```

Minion with lining numbers.

```
6696 ⟨*pmn⟩
6697 \SetProtrusion
6698    [ name      = pmnx-OT1,
6699      load      = pmnj-default ]
6700    { encoding = OT1,
6701      family   = pmnx }
6702    {
6703      1 = {230,180}
6704    }
6705
6706 \SetProtrusion
6707    [ name      = pmnx-T1,
6708      load      = pmnj-T1  ]
6709    { encoding = {T1,LY1},
6710      family   = pmnx      }
6711    {
6712      1 = {230,180}
6713    }
6714
6715 \SetProtrusion
6716    [ name      = pmnx-T2A,
6717      load      = pmnj-T2A  ]
6718    { encoding = {T2A},
6719      family   = pmnx      }
6720    {
6721      1 = {230,180}
6722    }
6723
6724 ⟨/pmn⟩
```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```
6725 ⟨*ptm⟩
6726 \SetProtrusion
6727    [ name      = ptm-LY1,
```

```
6728      load    = ptm-T1 ]
6729    { encoding = LY1,
6730      family  = {ptm,ptmx,ptmj} }
6731    {
6732      _                          = {100,100},
6733      \texttrademark             = {100,100},
6734      \textregistered            = {100,100},
6735      \textcopyright             = {100,100},
6736      \textdegree                = {300,300},
6737      \textminus                 = {200,200},
6738      \textellipsis              = {150,200},
6739 %    \texteuro                  = {    ,    }, % ?
6740      \textcent                  = {100,100},
6741      \textquotesingle           = {500,500},
6742      \textflorin                = { 50, 70},
6743      \textdagger                = {150,150},
6744      \textdaggerdbl             = {100,100},
6745      \textperthousand           = {    , 50},
6746      \textbullet                = {150,150},
6747      \textonesuperior           = {100,100},
6748      \texttwosuperior           = { 50, 50},
6749      \textthreesuperior         = { 50, 50},
6750      \textperiodcentered        = {300,300},
6751      \textplusminus             = { 50, 80},
6752      \textmultiply              = {100,100},
6753      \textdivide                = { 50,150}
```

Remaining slots in the source file.

```
6754    }
6755
6756 ⟨/ptm⟩
```

For the Greek LGR encoding.

```
6757 ⟨*ebg⟩
6758 \SetProtrusion
6759   [ name = EBGaramond-LGR ]
6760   { }
6761   {
6762     A = {50,50},
6763     D = {100,100},
6764     F = {50,50},
6765     G = {  ,150},
6766     K = {  ,50},
6767     L = {100,100},
6768     O = {50,50},
6769     U = {100,100},
6770     T = {50,50},
6771     W = {  ,50},
6772     Y = {50,50},
6773     . = { ,600},
6774   {,}= { ,500},
6775     : = { ,400},
6776     ; = { ,300},
6777     ! = { ,100},
6778     ? = { ,100},
6779     ~ = {200,250},
6780    \% = {50,50},
6781     * = {300,300},
6782     + = {250,250},
6783   {=}= { 50, 50},
6784     ( = {100,   },    ) = {    ,200},
6785     / = {100,200},
6786     - = {300,500},
6787    \texteuro = { 50,100},
6788    \textendash      = {300,300},   \textemdash        = {200,200},
```

```
6789      \textquoteleft    = {300,500},   \textquoteright   = {400,400},
6790      \guillemotleft    = {300,300},   \guillemotright   = {200,400},
6791    }
6792
6793 \SetProtrusion
6794    [ name     = EBGaramond-LGR-LF,
6795      load     = EBGaramond-LGR ]
6796    { encoding = LGR,
6797      family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
6798    {
6799      1 = {50,50},
6800      2 = {50,50},
6801      4 = {50,50},
6802      7 = {50,50},
6803    }
6804
6805 \SetProtrusion
6806    [ name     = EBGaramond-LGR-TOsF,
6807      load     = EBGaramond-LGR ]
6808    { encoding = LGR,
6809      family   = {EBGaramond-TOsF} }
6810    {
6811      1 = {150,150},
6812      2 = {50,50},
6813      3 = {50,50},
6814      4 = {50,50},
6815      5 = {50,50},
6816      6 = {50,50},
6817      7 = {50,80},
6818      8 = {50,50},
6819      9 = {50,50},
6820    }
6821
6822 ⟨/ebg⟩
```

2.8.2 Italics

To find default settings for italic is difficult, since the character shapes and their behaviour at the beginning or end of line may be wildly different for different fonts. In the generic settings we therefore omit the letters, and only set up the punctuation characters.

The italic glyphs of Computer Modern Roman feature a lot of side bearing, therefore almost all of them have to protrude. [12]

```
6823 \SetProtrusion
6824 ⟨m-t⟩    [ name     = OT1-it   ]
6825 ⟨bch⟩    [ name     = bch-it   ]
6826 ⟨blg⟩    [ name     = blg-it,
6827 ⟨blg⟩      load     = blg-default ]
6828 ⟨cmr⟩    [ name     = cmr-it   ]
6829 ⟨ebg⟩    [ name     = EBGaramond-it   ]
6830 ⟨pmn⟩    [ name     = pmnj-it  ]
6831 ⟨ppl⟩    [ name     = ppl-it   ]
6832 ⟨ptm⟩    [ name     = ptm-it   ]
6833 ⟨ugm⟩    [ name     = ugm-it   ]
6834 ⟨m-t|bch|blg|ugm⟩   { encoding = OT1,
6835 ⟨ppl|ptm⟩   { encoding = {OT1,OT4},
6836 ⟨bch⟩      family   = bch,
6837 ⟨blg⟩      family   = blg,
6838 ⟨ppl⟩      family   = {ppl,pplx,pplj},
6839 ⟨ptm⟩      family   = {ptm,ptmx,ptmj},
```

---

12   Settings contributed by *Hendrik Vogt*.

```
6840 ⟨ugm⟩      family   = ugm,
6841 ⟨m-t|bch|ppl|ptm⟩     shape    = {it,sl}  }
6842 ⟨blg|ugm⟩    shape    = it   }
6843 ⟨cmr|ebg|pmn⟩   { }
6844    {
6845 ⟨cmr⟩      A = {100,100},
6846 ⟨ptm⟩      A = {100,50},
6847 ⟨ebg|pmn⟩     A = {50,   },
6848 ⟨ugm⟩      A = {   ,150},
6849 ⟨ppl⟩      A = {50,50},
6850 ⟨ptm⟩     \AE = {100,   },
6851 ⟨ebg|ppl⟩    \AE = {50,  },
6852 ⟨cmr⟩      B = {83,-40},
6853 ⟨ebg|ppl|ptm⟩      B = {50,   },
6854 ⟨pmn⟩      B = {20,-50},
6855 ⟨bch|ppl|ptm|ugm⟩     C = {50,   },
6856 ⟨cmr⟩      C = {165,-75},
6857 ⟨ebg⟩      C = {100,  },
6858 ⟨pmn⟩      C = {50,-50},
6859 ⟨cmr⟩      D = {75, -28},
6860 ⟨ebg|ppl|ptm⟩      D = {50,50},
6861 ⟨pmn⟩      D = {20,   },
6862 ⟨cmr⟩      E = {80,-55},
6863 ⟨ebg|ppl|ptm⟩      E = {50,   },
6864 ⟨pmn⟩      E = {20,-50},
6865 ⟨cmr⟩      F = {85,-80},
6866 ⟨ebg|ptm⟩      F = {100,  },
6867 ⟨pmn⟩      F = {10,   },
6868 ⟨ppl⟩      F = {50,   },
6869 ⟨bch|ppl|ptm|ugm⟩     G = {50,   },
6870 ⟨cmr⟩      G = {153,-15},
6871 ⟨ebg⟩      G = {100,  },
6872 ⟨pmn⟩      G = {50,-50},
6873 ⟨cmr⟩      H = {73,-60},
6874 ⟨ebg|ppl|ptm⟩      H = {50,   },
6875 ⟨cmr⟩      I = {140,-120},
6876 ⟨ebg|ptm⟩      I = {50,   },
6877 ⟨pmn⟩      I = {20,-50},
6878 ⟨cmr⟩      J = {135,-80},
6879 ⟨ebg⟩      J = {50,   },
6880 ⟨pmn⟩      J = {20,   },
6881 ⟨ptm⟩      J = {100,  },
6882 ⟨cmr⟩      K = {70,-30},
6883 ⟨ebg|ppl|ptm⟩      K = {50,   },
6884 ⟨pmn⟩      K = {20,   },
6885 ⟨cmr⟩      L = {87, 40},
6886 ⟨ebg|ppl|ptm⟩      L = {50,   },
6887 ⟨pmn⟩      L = {20,50},
6888 ⟨ugm⟩      L = {   ,100},
6889 ⟨cmr⟩      M = {67,-45},
6890 ⟨pmn⟩      M = {   ,-30},
6891 ⟨ptm⟩      M = {50,   },
6892 ⟨cmr⟩      N = {75,-55},
6893 ⟨pmn⟩      N = {   ,-30},
6894 ⟨ptm⟩      N = {50,   },
6895 ⟨bch|pmn|ppl|ptm⟩      O = {50,   },
6896 ⟨cmr⟩      O = {150,-30},
6897 ⟨ebg⟩      O = {100,  },
6898 ⟨ugm⟩      O = {70,50},
6899 ⟨ppl|ptm⟩    \OE = {50,   },
6900 ⟨ebg⟩     \OE = {100,  },
6901 ⟨cmr⟩      P = {82,-50},
6902 ⟨ebg|ppl|ptm⟩      P = {50,   },
6903 ⟨pmn⟩      P = {20,-50},
6904 ⟨bch|pmn|ppl|ptm⟩      Q = {50,   },
```

```
6905 ⟨cmr⟩      Q = {150,-30},
6906 ⟨ebg⟩      Q = {100,  },
6907 ⟨ugm⟩      Q = {70,50},
6908 ⟨cmr⟩      R = {75, 15},
6909 ⟨ebg|ppl|ptm⟩      R = {50,  },
6910 ⟨pmn⟩      R = {20,  },
6911 ⟨bch|ebg|ppl|ptm⟩      S = {50,  },
6912 ⟨cmr⟩      S = {90,-65},
6913 ⟨pmn⟩      S = {20,-30},
6914 ⟨bch|ebg|ppl|ptm⟩      $ = {50,  },
6915 ⟨cmr⟩      $ = {100,-20},
6916 ⟨pmn⟩      $ = {20,-30},
6917 ⟨bch|pmn|ugm⟩      T = {70,  },
6918 ⟨cmr⟩      T = {220,-85},
6919 ⟨ebg|ppl|ptm⟩      T = {100,  },
6920 ⟨cmr⟩      U = {230,-55},
6921 ⟨ebg|ppl|ptm⟩      U = {50,  },
6922 ⟨pmn⟩      U = {50,-50},
6923 ⟨cmr⟩      V = {260,-60},
6924 ⟨ebg|pmn|ugm⟩      V = {100,  },
6925 ⟨ppl|ptm⟩      V = {100,50},
6926 ⟨cmr⟩      W = {185,-55},
6927 ⟨ebg|pmn|ugm⟩      W = {100,  },
6928 ⟨ppl⟩      W = {50,  },
6929 ⟨ptm⟩      W = {100,50},
6930 ⟨cmr⟩      X = {70,-30},
6931 ⟨ppl|ptm⟩      X = {50,  },
6932 ⟨cmr⟩      Y = {250,-60},
6933 ⟨pmn⟩      Y = {50,  },
6934 ⟨ppl⟩      Y = {100,50},
6935 ⟨ptm⟩      Y = {100,  },
6936 ⟨cmr⟩      Z = {90,-60},
6937 ⟨pmn⟩      Z = {  ,-50},
6938 ⟨cmr⟩      a = {150,-10},
6939 ⟨cmr⟩      b = {170,  },
6940 ⟨cmr⟩      c = {173,-10},
6941 ⟨cmr⟩      d = {150,-55},
6942 ⟨pmn⟩      d = {  ,-50},
6943 ⟨cmr⟩      e = {180,  },
6944 ⟨cmr⟩      f = {  ,-250},
6945 ⟨ebg|pmn⟩      f = {  ,-100},
6946 ⟨cmr⟩      g = {150,-10},
6947 ⟨cmr⟩      h = {100,  },
6948 ⟨cmr⟩      i = {210,  },
6949 ⟨pmn⟩      i = {  ,-30},
6950 ⟨cmr⟩      j = {  ,-40},
6951 ⟨pmn⟩      j = {  ,-30},
6952 ⟨cmr⟩      k = {110,-50},
6953 ⟨cmr⟩      l = {240,-110},
6954 ⟨pmn⟩      l = {  ,-100},
6955 ⟨cmr⟩      m = {80,  },
6956 ⟨cmr⟩      n = {115,  },
6957 ⟨bch⟩      o = {50,50},
6958 ⟨cmr⟩      o = {155,  },
6959 ⟨bch⟩      p = {  ,50},
6960 ⟨pmn⟩      p = {-50,  },
6961 ⟨bch⟩      q = {50,  },
6962 ⟨cmr⟩      q = {170,-40},
6963 ⟨cmr⟩      r = {155,-40},
6964 ⟨pmn⟩      r = {  ,50},
6965 ⟨cmr⟩      s = {130,  },
6966 ⟨bch⟩      t = {  ,50},
6967 ⟨cmr⟩      t = {230,-10},
6968 ⟨cmr⟩      u = {120,  },
6969 ⟨cmr⟩      v = {140,-25},
```

```
6970 ⟨pmn|ugm⟩     v = {50,   },
6971 ⟨bch⟩      w = {  ,50},
6972 ⟨cmr⟩      w = {98,-20},
6973 ⟨pmn|ugm⟩     w = {50,   },
6974 ⟨cmr⟩      x = {65,-40},
6975 ⟨bch⟩      y = {  ,50},
6976 ⟨cmr⟩      y = {130,-20},
6977 ⟨cmr⟩      z = {110,-80},
6978 ⟨cmr⟩      0 = {170,-85},
6979 ⟨bch|ptm⟩    1 = {150,100},
6980 ⟨cmr⟩      1 = {230,110},
6981 ⟨ebg⟩      1 = {150,   },
6982 ⟨pmn⟩      1 = {50,   },
6983 ⟨ppl⟩      1 = {100,   },
6984 ⟨ugm⟩      1 = {150,150},
6985 ⟨cmr⟩      2 = {130,-70},
6986 ⟨ebg|ppl|ptm⟩    2 = {50,   },
6987 ⟨pmn⟩      2 = {-50,   },
6988 ⟨bch⟩      3 = {50,   },
6989 ⟨cmr⟩      3 = {140,-70},
6990 ⟨pmn⟩      3 = {-100,   },
6991 ⟨ptm⟩      3 = {100,50},
6992 ⟨bch⟩      4 = {100,   },
6993 ⟨cmr⟩      4 = {130,80},
6994 ⟨ebg⟩      4 = {150,   },
6995 ⟨ppl|ptm⟩     4 = {50,   },
6996 ⟨cmr⟩      5 = {160,   },
6997 ⟨ptm⟩      5 = {50,   },
6998 ⟨bch⟩      6 = {50,   },
6999 ⟨cmr⟩      6 = {175,-30},
7000 ⟨bch|ebg|ptm⟩    7 = {100,   },
7001 ⟨cmr⟩      7 = {250,-150},
7002 ⟨pmn⟩      7 = {20,   },
7003 ⟨ppl⟩      7 = {50,   },
7004 ⟨cmr⟩      8 = {130,-40},
7005 ⟨cmr⟩      9 = {155,-80},
7006 ⟨m-t|cmr|ebg|pmn|ppl⟩    . = {  ,500},
7007 ⟨blg⟩      . = {400,600},
7008 ⟨bch|ptm|ugm⟩    . = {  ,700},
7009 ⟨blg⟩     {,}= {300,500},
7010 ⟨m-t|ebg|pmn|ppl⟩   {,}= {  ,500},
7011 ⟨cmr⟩     {,}= {  ,450},
7012 ⟨bch|ugm⟩    {,}= {  ,600},
7013 ⟨ptm⟩     {,}= {  ,700},
7014 ⟨m-t|cmr|ebg|ppl⟩    : = {  ,300},
7015 ⟨bch|ugm⟩    : = {  ,400},
7016 ⟨pmn⟩      : = {  ,200},
7017 ⟨ptm⟩      : = {  ,500},
7018 ⟨m-t|cmr|ebg|ppl⟩    ; = {  ,300},
7019 ⟨bch|ugm⟩    ; = {  ,400},
7020 ⟨pmn⟩      ; = {  ,200},
7021 ⟨ptm⟩      ; = {  ,500},
7022 ⟨ptm⟩      ! = {  ,100},
7023 ⟨bch⟩      ? = {  ,200},
7024 ⟨ptm⟩      ? = {  ,100},
7025 ⟨ppl⟩      ? = {  ,300},
7026 ⟨pmn⟩      " = {400,200},
7027 ⟨m-t|ebg|pmn|ppl|ptm⟩    & = {50,50},
7028 ⟨bch⟩      & = {  ,80},
7029 ⟨cmr⟩      & = {130,30},
7030 ⟨ugm⟩      & = {50,100},
7031 ⟨m-t|ebg|pmn⟩    \% = {100,   },
7032 ⟨cmr⟩      \% = {180,50},
7033 ⟨bch⟩      \% = {50,50},
7034 ⟨ppl|ptm⟩     \% = {100,100},
```

```
7035 ⟨ugm⟩    \% = {100,50},
7036 ⟨m-t|pmn|ppl⟩     * = {200,200},
7037 ⟨bch⟩     * = {300,200},
7038 ⟨cmr⟩     * = {380,20},
7039 ⟨ebg⟩     * = {500,100},
7040 ⟨ptm|ugm⟩     * = {400,200},
7041 ⟨m-t|pmn|ppl⟩     + = {150,200},
7042 ⟨cmr⟩     + = {180,200},
7043 ⟨bch|ugm⟩     + = {250,250},
7044 ⟨ebg|ptm⟩     + = {250,200},
7045 ⟨m-t|ebg|pmn|ppl⟩     @ = {50,50},
7046 ⟨bch⟩     @ = {80,50},
7047 ⟨cmr⟩     @ = {180,10},
7048 ⟨ptm⟩     @ = {150,150},
7049 ⟨m-t|bch|ugm⟩     ~ = {150,150},
7050 ⟨cmr|ebg|pmn|ppl|ptm⟩     ~ = {200,150},
7051 ⟨ugm⟩     {=}= {200,200},
7052 ⟨m-t|bch|ebg|pmn|ppl|ptm|ugm⟩     ( = {200, },     ) = {  ,200},
7053 ⟨cmr⟩     ( = {300, },     ) = {  ,70},
7054 ⟨m-t|ebg|ppl|ptm|ugm⟩     / = {100,200},
7055 ⟨cmr⟩     / = {100,100},
7056 ⟨bch⟩     / = {  ,150},
7057 ⟨pmn⟩     / = {100,150},
7058 ⟨m-t⟩     - = {300,300},
7059 ⟨bch|ebg⟩     - = {300,400},
7060 ⟨pmn⟩     - = {200,300},
7061 ⟨cmr⟩     - = {500,300},
7062 ⟨ppl⟩     - = {300,500},
7063 ⟨ptm⟩     - = {500,500},
7064 ⟨ugm⟩     - = {400,700},
7065 ⟨blg⟩     _ = {0,300},
7066 ⟨m-t|pmn⟩     \textendash     = {200,200},   \textemdash     = {150,150},
7067 ⟨bch⟩     \textendash     = {200,300},   \textemdash     = {150,200},
7068 ⟨cmr⟩     \textendash     = {500,300},   \textemdash     = {400,170},
7069 ⟨ebg|ppl|ptm|ugm⟩     \textendash     = {300,300},   \textemdash     = {200,200},
7070 ⟨m-t|bch|pmn|ugm⟩     \textquoteleft     = {400,200},   \textquoteright     = {400,200},
7071 ⟨blg⟩     \textquoteleft     = {400,400},   \textquoteright     = {400,400},
7072 ⟨cmr⟩     \textquoteleft     = {800,200},   \textquoteright     = {800,-20},
7073 ⟨ebg⟩     \textquoteleft     = {800,200},   \textquoteright     = {800,200},
7074 ⟨ppl⟩     \textquoteleft     = {700,400},   \textquoteright     = {700,400},
7075 ⟨ptm⟩     \textquoteleft     = {800,500},   \textquoteright     = {800,500},
7076 ⟨m-t|bch|pmn⟩     \textquotedblleft = {400,200},   \textquotedblright = {400,200}
7077 ⟨blg⟩     \textquotedblright = {300,300}
7078 ⟨cmr⟩     \textquotedblleft = {540,100},   \textquotedblright = {500,100}
7079 ⟨ebg⟩     \textquotedblleft = {700,200},   \textquotedblright = {700,200}
7080 ⟨ppl⟩     \textquotedblleft = {500,300},   \textquotedblright = {500,300}
7081 ⟨ptm⟩     \textquotedblleft = {700,400},   \textquotedblright = {700,400}
7082 ⟨ugm⟩     \textquotedblleft = {600,200},   \textquotedblright = {600,200}
7083     }
7084
7085 ⟨*cmr|ebg|pmn⟩
7086 \SetProtrusion
7087 ⟨cmr⟩   [ name    = cmr-it-OT1,
7088 ⟨ebg⟩   [ name    = EBGaramond-it-OT1,
7089 ⟨pmn⟩   [ name    = pmnj-it-OT1,
7090 ⟨cmr⟩     load    = cmr-it   ]
7091 ⟨ebg⟩     load    = EBGaramond-it   ]
7092 ⟨pmn⟩     load    = pmnj-it  ]
7093 ⟨cmr⟩   { encoding = {OT1,OT4},
7094 ⟨pmn⟩   { encoding = OT1,
7095 ⟨cmr⟩     family  = cmr,
7096 ⟨pmn⟩     family  = pmnj,
7097 ⟨cmr⟩     shape   = it       }
7098 ⟨pmn⟩     shape   = {it,sl}  }
7099 ⟨ebg⟩   { }
```

```
7100      {
7101 ⟨cmr⟩     \AE = {100,    },
7102 ⟨pmn⟩     \AE = {    ,-50},
7103 ⟨cmr⟩     \OE = {100,    },
7104 ⟨pmn⟩     \OE = {50,    }
7105 ⟨*cmr|ebg⟩
7106 ⟨cmr⟩     "00 = {200,150}, % \Gamma
7107 ⟨ebg⟩     "00 = {    ,150}, % \Gamma
7108 ⟨cmr⟩     "01 = {150,100}, % \Delta
7109 ⟨ebg⟩     "01 = {100,100}, % \Delta
7110 ⟨cmr⟩     "02 = {150, 50}, % \Theta
7111 ⟨ebg⟩     "02 = { 50, 50}, % \Theta
7112 ⟨cmr⟩     "03 = {150, 50}, % \Lambda
7113 ⟨ebg⟩     "03 = {100,100}, % \Lambda
7114 ⟨cmr⟩     "04 = {100,100}, % \Xi
7115 ⟨ebg⟩     "04 = { 50, 50}, % \Xi
7116 ⟨cmr⟩     "05 = {100,100}, % \Pi
7117 ⟨cmr⟩     "06 = {100, 50}, % \Sigma
7118 ⟨cmr⟩     "07 = {200,150}, % \Upsilon
7119 ⟨ebg⟩     "07 = {100,100}, % \Upsilon
7120 ⟨cmr⟩     "08 = {150, 50}, % \Phi
7121 ⟨ebg⟩     "08 = { 50, 50}, % \Phi
7122 ⟨cmr⟩     "09 = {150,100}, % \Psi
7123 ⟨ebg⟩     "09 = { 50, 50}, % \Psi
7124      "0A = { 50, 50}, % \Omega
7125 ⟨ebg⟩     138 = {    , 50}, % \L
7126 ⟨/cmr|ebg⟩
7127      }
7128
7129 ⟨/cmr|ebg|pmn⟩
7130 ⟨*ebg⟩
7131 \SetProtrusion
7132    [ name     = EBGaramond-it-OT1-LF,
7133      load     = EBGaramond-it-OT1 ]
7134    { encoding = OT1,
7135      family   = {EBGaramond-LF,EBGaramond-TLF},
7136      shape    = it }
7137    {
7138      1 = {50,50},
7139      2 = {50,50},
7140      3 = {80,50},
7141      4 = {50,50},
7142      5 = {50,50},
7143      6 = {50,50},
7144      7 = {50,50},
7145      8 = {50,50},
7146      9 = {50,  },
7147    }
7148
7149 \SetProtrusion
7150    [ name     = EBGaramond-it-OT1-OsF,
7151      load     = EBGaramond-it-OT1 ]
7152    { encoding = OT1,
7153      family   = {EBGaramond-OsF},
7154      shape    = it }
7155    {
7156      1 = {50,50},
7157      2 = {50,50},
7158      3 = {  ,80},
7159      4 = {50,50},
7160      7 = {50,50},
7161    }
7162
7163 \SetProtrusion
7164    [ name     = EBGaramond-it-OT1-TOsF,
```

```
7165        load      = EBGaramond-it-OT1 ]
7166      { encoding = OT1,
7167        family    = {EBGaramond-TOsF},
7168        shape     = it }
7169      {
7170        0 = {150,150},
7171        1 = {150,150},
7172        2 = {80,80},
7173        3 = {50,80},
7174        4 = {50,80},
7175        5 = {50,80},
7176        6 = {50,50},
7177        7 = {50,100},
7178        8 = {50,50},
7179        9 = {50,80},
7180      }
7181
7182 ⟨/ebg⟩
7183 \SetProtrusion
7184 ⟨m-t⟩   [ name      = T1-it-default,
7185 ⟨bch⟩   [ name      = bch-it-T1,
7186 ⟨blg⟩   [ name      = blg-it-T1,
7187 ⟨cmr⟩   [ name      = cmr-it-T1,
7188 ⟨ebg⟩   [ name      = EBGaramond-it-T1,
7189 ⟨pmn⟩   [ name      = pmnj-it-T1,
7190 ⟨ppl⟩   [ name      = ppl-it-T1,
7191 ⟨ptm⟩   [ name      = ptm-it-T1,
7192 ⟨ugm⟩   [ name      = ugm-it-T1,
7193 ⟨m-t⟩     load      = OT1-it   ]
7194 ⟨bch⟩     load      = bch-it   ]
7195 ⟨blg⟩     load      = blg-T1   ]
7196 ⟨cmr⟩     load      = cmr-it   ]
7197 ⟨pmn⟩     load      = pmnj-it  ]
7198 ⟨ebg⟩     load      = EBGaramond-it   ]
7199 ⟨ppl⟩     load      = ppl-it   ]
7200 ⟨ptm⟩     load      = ptm-it   ]
7201 ⟨ugm⟩     load      = ugm-it   ]
7202 ⟨m-t|bch|cmr|pmn|ppl⟩   { encoding = {T1,LY1},
7203 ⟨ebg⟩     { encoding = {LY1},
7204 ⟨blg|ptm|ugm⟩   { encoding = T1,
7205 ⟨bch⟩     family    = bch,
7206 ⟨blg⟩     family    = blg,
7207 ⟨cmr⟩     family    = cmr,
7208 ⟨pmn⟩     family    = pmnj,
7209 ⟨ebg⟩     family    = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
7210 ⟨ppl⟩     family    = {ppl,pplx,pplj},
7211 ⟨ptm⟩     family    = {ptm,ptmx,ptmj},
7212 ⟨ugm⟩     family    = ugm,
7213 ⟨m-t|bch|pmn|ppl|ptm⟩     shape     = {it,sl}  }
7214 ⟨blg|cmr|ebg|ugm⟩         shape     = it        }
7215    {
7216 ⟨m-t|bch|pmn⟩      _ = {   ,100},
7217 ⟨blg⟩     _ = {0,300},
7218 ⟨cmr|ugm⟩     _ = {100,200},
7219 ⟨ebg|ppl|ptm⟩     _ = {100,100},
7220 ⟨blg⟩    . = {400,600},
7221 ⟨blg⟩    {,}= {300,500},
7222 ⟨cmr⟩    \AE = {100,    },
7223 ⟨pmn⟩    \AE = {   ,-50},
7224 ⟨bch|pmn⟩      \OE = { 50,    },
7225 ⟨cmr⟩    \OE = {100,    },
7226 ⟨pmn⟩     031 = {  ,-100}, % ffl
7227 ⟨cmr|ptm⟩      156 = {100, },  % IJ
7228 ⟨ebg⟩     156 = {50,  },  % IJ
7229 ⟨pmn⟩     156 = {20,  },  % IJ
```

```
7230 ⟨pmn⟩     188 = {   ,-30}, % ij
7231 ⟨pmn⟩   \v t = { ,100},
7232 ⟨m-t|ebg|ppl|ptm⟩     \textbackslash    = {100,200},
7233 ⟨cmr|ugm⟩   \textbackslash    = {300,300},
7234 ⟨bch⟩   \textbackslash    = {150,150},
7235 ⟨pmn⟩    \textbackslash    = {100,150},
7236 ⟨ugm⟩    \textbar          = {200,200},
7237 ⟨cmr⟩    \textquotedblleft = {500,300},
7238 ⟨blg⟩   \textquoteleft    = {400,400},    \textquoteright    = {400,400},
7239 ⟨blg⟩   \textquotedbl     = {300,300},    \textquotedblleft  = {300,300},
7240 ⟨blg⟩   \textquotedblright = {300,300},    \quotedblbase      = {200,600},
7241 ⟨m-t|ptm⟩    \quotesinglbase   = {300,700},    \quotedblbase      = {400,500},
7242 ⟨cmr⟩    \quotesinglbase   = {300,700},    \quotedblbase      = {200,600},
7243 ⟨bch|pmn⟩    \quotesinglbase   = {200,500},    \quotedblbase      = {150,500},
7244 ⟨ebg|ppl⟩    \quotesinglbase   = {500,500},    \quotedblbase      = {400,400},
7245 ⟨ugm⟩    \quotesinglbase   = {300,700},    \quotedblbase      = {300,500},
7246 ⟨m-t|ppl|ptm⟩    \guilsinglleft    = {400,400},    \guilsinglright    = {300,500},
7247 ⟨bch|pmn⟩    \guilsinglleft    = {300,400},    \guilsinglright    = {200,500},
7248 ⟨cmr⟩    \guilsinglleft    = {500,300},    \guilsinglright    = {400,400},
7249 ⟨ebg⟩    \guilsinglleft    = {500,400},    \guilsinglright    = {300,500},
7250 ⟨ugm⟩    \guilsinglleft    = {400,400},    \guilsinglright    = {300,600},
7251 ⟨m-t|ppl⟩    \guillemotleft    = {300,300},    \guillemotright    = {300,300},
7252 ⟨bch|pmn⟩    \guillemotleft    = {200,300},    \guillemotright    = {150,400},
7253 ⟨cmr⟩    \guillemotleft    = {400,100},    \guillemotright    = {200,300},
7254 ⟨ebg⟩    \guillemotleft    = {300,300},    \guillemotright    = {200,400},
7255 ⟨ptm⟩    \guillemotleft    = {300,400},    \guillemotright    = {200,400},
7256 ⟨ugm⟩    \guillemotleft    = {300,400},    \guillemotright    = {300,400},
7257 ⟨m-t|ebg|ppl|ugm⟩     \textexclamdown    = {100,   },    \textquestiondown = {200,   },
7258 ⟨cmr|ptm⟩    \textexclamdown    = {200,   },    \textquestiondown = {200,   },
7259 ⟨pmn⟩    \textexclamdown    = {-50,   },    \textquestiondown = {-50,   },
7260 ⟨m-t|ppl|ugm⟩    \textbraceleft     = {200,100},    \textbraceright    = {200,200},
7261 ⟨bch|pmn⟩    \textbraceleft     = {200,   },    \textbraceright    = {   ,200},
7262 ⟨cmr|ebg|ptm⟩     \textbraceleft     = {400,100},    \textbraceright    = {200,200},
7263 ⟨bch|pmn⟩    \textless          = {100,   },    \textgreater      = {   ,100},
7264 ⟨cmr|ebg|ppl|ptm⟩     \textless          = {300,100},    \textgreater      = {200,100}
7265 ⟨pmn⟩    \textvisiblespace = {100,100}
7266  }
7267
7268 ⟨*ebg⟩
7269 \SetProtrusion
7270   [ name     = EBGaramond-it-T1-LF,
7271     load     = EBGaramond-it-T1 ]
7272   { encoding = T1,
7273     family   = {EBGaramond-LF,EBGaramond-TLF},
7274     shape    = it }
7275   {
7276     1 = {50,50},
7277     2 = {50,50},
7278     3 = {80,50},
7279     4 = {50,50},
7280     5 = {50,50},
7281     6 = {50,50},
7282     7 = {50,50},
7283     8 = {50,50},
7284     9 = {50,  },
7285   }
7286
7287 \SetProtrusion
7288   [ name     = EBGaramond-it-T1-OsF,
7289     load     = EBGaramond-it-T1 ]
7290   { encoding = T1,
7291     family   = {EBGaramond-OsF},
7292     shape    = it }
7293   {
7294     1 = {50,50},
```

```
7295        2 = {50,50},
7296        3 = {  ,80},
7297        4 = {50,50},
7298        7 = {50,50},
7299      }
7300
7301  \SetProtrusion
7302     [ name     = EBGaramond-it-T1-TOsF,
7303       load     = EBGaramond-it-T1 ]
7304     { encoding = T1,
7305       family   = {EBGaramond-TOsF},
7306       shape    = it }
7307     {
7308       0 = {150,150},
7309       1 = {150,150},
7310       2 = {80,80},
7311       3 = {50,80},
7312       4 = {50,80},
7313       5 = {50,80},
7314       6 = {50,50},
7315       7 = {50,100},
7316       8 = {50,50},
7317       9 = {50,80},
7318     }
7319
7320  ⟨/ebg⟩
7321  ⟨*m-t|cmr|pmn⟩
7322  \SetProtrusion
7323  ⟨m-t⟩    [ name     = T2A-it-default,
7324  ⟨cmr⟩    [ name     = cmr-it-T2A,
7325  ⟨pmn⟩    [ name     = pmnj-it-T2A,
7326  ⟨m-t⟩      load     = OT1-it   ]
7327  ⟨cmr⟩      load     = cmr-it   ]
7328  ⟨pmn⟩      load     = pmnj-it  ]
7329     { encoding = T2A,
7330  ⟨cmr⟩      family   = cmr,
7331  ⟨pmn⟩      family   = pmnj,
7332  ⟨m-t|pmn⟩    shape    = {it,sl}  }
7333  ⟨cmr⟩      shape    = it         }
7334     {
7335  ⟨cmr⟩      \CYRA = {100,50},
7336  ⟨pmn⟩      \CYRA = {50,  },
7337  ⟨cmr⟩      \CYRB = {50,  },
7338  ⟨cmr⟩      \CYRV = {50,  },
7339  ⟨pmn⟩      \CYRV = {20,-50},
7340  ⟨cmr⟩      \CYRG = {100, },
7341  ⟨pmn⟩      \CYRG = {10,  },
7342  ⟨cmr⟩      \CYRD = {50,  },
7343  ⟨cmr⟩      \CYRE = {50,  },
7344  ⟨pmn⟩      \CYRE = {20,-50},
7345  ⟨cmr⟩      \CYRZH = {50,  },
7346  ⟨cmr⟩      \CYRZ = {50,  },
7347  ⟨pmn⟩      \CYRZ = {20,-50},
7348  ⟨cmr⟩      \CYRI = {50,  },
7349  ⟨pmn⟩      \CYRI = {  ,-30},
7350  ⟨cmr⟩      \CYRISHRT = {50,  },
7351  ⟨cmr⟩      \CYRK = {50,  },
7352  ⟨pmn⟩      \CYRK = {20,  },
7353  ⟨cmr⟩      \CYRL = {50,  },
7354  ⟨cmr⟩      \CYRM = {50,  },
7355  ⟨pmn⟩      \CYRM = {  ,-30},
7356  ⟨cmr⟩      \CYRN = {50,  },
7357  ⟨cmr⟩      \CYRO = {100, },
7358  ⟨pmn⟩      \CYRO = {50,  },
7359  ⟨cmr⟩      \CYRP = {50,  },
```

```
7360 ⟨cmr⟩      \CYRR = {50,  },
7361 ⟨pmn⟩      \CYRR = {20,-50},
7362 ⟨cmr⟩      \CYRS = {100, },
7363 ⟨pmn⟩      \CYRS = {50,  },
7364 ⟨cmr⟩      \CYRT = {100, },
7365 ⟨pmn⟩      \CYRT = {70,  },
7366 ⟨cmr⟩      \CYRU = {100, },
7367 ⟨pmn⟩      \CYRU = {50,  },
7368 ⟨cmr⟩      \CYRF = {100, },
7369 ⟨cmr⟩      \CYRH = {50,  },
7370 ⟨cmr⟩      \CYRC = {50,  },
7371 ⟨cmr⟩      \CYRCH = {100, },
7372 ⟨cmr⟩      \CYRSH = {50,  },
7373 ⟨cmr⟩      \CYRSHCH = {50,  },
7374 ⟨cmr⟩      \CYRHRDSN = {100, },
7375 ⟨cmr⟩      \CYRERY = {50,  },
7376 ⟨cmr⟩      \CYRSFTSN = {50,  },
7377 ⟨cmr⟩      \CYREREV = {50,  },
7378 ⟨cmr⟩      \CYRYU = {50,  },
7379 ⟨cmr⟩      \CYRYA = {50,  },
7380 ⟨pmn⟩      \CYRYA = {  ,20},
7381 ⟨pmn⟩      \cyrr = {-50, },
7382 ⟨m-t|pmn⟩      _ = {  ,100},
7383 ⟨cmr⟩        _ = {100,200},
7384 ⟨pmn⟩      031 = { ,-100}, % ffl
7385 ⟨pmn⟩      \v t = { ,100},
7386 ⟨m-t⟩     \textbackslash    = {100,200},   \quotedblbase    = {400,500},
7387 ⟨cmr⟩     \textbackslash    = {300,300},   \quotedblbase    = {200,600},
7388 ⟨pmn⟩     \textbackslash    = {100,150},   \quotedblbase    = {150,500},
7389 ⟨m-t⟩     \guillemotleft    = {300,300},   \guillemotright  = {300,300},
7390 ⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright  = {200,300},
7391 ⟨pmn⟩     \guillemotleft    = {200,300},   \guillemotright  = {150,400},
7392 ⟨m-t⟩     \textbraceleft    = {200,100},   \textbraceright  = {200,200},
7393 ⟨cmr⟩     \textbraceleft    = {400,100},   \textbraceright  = {200,200},
7394 ⟨pmn⟩     \textbraceleft    = {200, },     \textbraceright  = {  ,200},
7395 ⟨cmr⟩     \textquotedblleft = {500,300},
7396 ⟨cmr⟩     \textless         = {300,100},   \textgreater     = {200,100}
7397 ⟨pmn⟩     \textless         = {100, },     \textgreater     = {  ,100}
7398    }
7399
7400 ⟨/m-t|cmr|pmn⟩
7401 ⟨*m-t|ptm⟩
7402 \SetProtrusion
7403 ⟨m-t⟩   [ name     = QX-it-default,
7404 ⟨ptm⟩   [ name     = ptm-it-QX,
7405 ⟨m-t⟩     load     = OT1-it ]
7406 ⟨ptm⟩     load     = ptm-it ]
7407    { encoding = {QX},
7408 ⟨ptm⟩     family   = {ptm,ptmx,ptmj},
7409     shape    = {it,sl} }
7410    {
7411 ⟨ptm⟩     009 = {  , 50}, % fk
7412    {=} = {100,100},
7413 ⟨m-t⟩     \textunderscore   = {100,100},
7414 ⟨ptm⟩     \textunderscore   = {100,150},
7415    \textbackslash    = {100,200},
7416    \quotedblbase     = {300,400},
7417 ⟨m-t⟩     \guillemotleft    = {300,300},   \guillemotright     = {300,300},
7418 ⟨ptm⟩     \guillemotleft    = {200,400},   \guillemotright     = {200,400},
7419    \textexclamdown   = {200, },     \textquestiondown = {200, },
7420    \textbraceleft    = {200,100},   \textbraceright   = {200,200},
7421    \textless         = {100,100},   \textgreater      = {100,100},
7422    \textminus        = {200,200},   \textdegree       = {300,150},
7423 ⟨m-t⟩     \copyright        = {100,100},   \textregistered   = {100,100}
7424 ⟨ptm⟩     \textregistered   = {100,150},   \copyright        = {100,150},
```

```
7425 ⟨ptm⟩    \textDelta      = { 70,   },   \textdelta        = {   , 50},
7426 ⟨ptm⟩    \textpi         = { 50, 80},   \textmu           = {   , 80},
7427 ⟨ptm⟩    \texteuro       = {200,   },   \textellipsis     = {100,200},
7428 ⟨ptm⟩    \textquoteleft  = {500,400},   \textquoteright   = {500,400},
7429 ⟨ptm⟩    \textquotedblleft = {500,300}, \textquotedblright = {400,400},
7430 ⟨ptm⟩    \textapprox     = { 50, 50},   \textinfty        = {100,100},
7431 ⟨ptm⟩    \textdagger     = {150,150},   \textdaggerdbl    = {100,100},
7432 ⟨ptm⟩    \textdiv        = {150,150},   \textasciitilde   = { 80, 80},
7433 ⟨ptm⟩    \texttimes      = {100,150},   \textpm           = { 50, 80},
7434 ⟨ptm⟩    \textbullet     = {300,100},   \textperiodcentered = {300,300},
7435 ⟨ptm⟩    \textquotesingle = {500,500},  \textquotedbl     = {300,300},
7436 ⟨ptm⟩    \textperthousand = {   ,50}
7437      }
7438
7439 ⟨/m-t|ptm⟩
7440 ⟨*cmr|bch⟩
7441 \SetProtrusion
7442 ⟨cmr⟩   [ name = cmr-it-T5,
7443 ⟨cmr⟩     load = cmr-it ]
7444 ⟨bch⟩   [ name = bch-it-T5,
7445 ⟨bch⟩     load = bch-it ]
7446   { encoding = T5,
7447 ⟨bch⟩     family  = bch,
7448 ⟨cmr⟩     family  = cmr,
7449     shape   = it }
7450   {
7451 ⟨bch⟩     _ = {   ,100},
7452 ⟨cmr⟩     _ = {100,200},
7453 ⟨bch⟩     \textbackslash    = {150,150},
7454 ⟨cmr⟩     \textbackslash    = {300,300},
7455 ⟨bch⟩     \quotesinglbase   = {200,500},   \quotedblbase     = {150,500},
7456 ⟨cmr⟩     \quotesinglbase   = {300,700},   \quotedblbase     = {200,600},
7457 ⟨bch⟩     \guilsinglleft    = {300,400},   \guilsinglright   = {200,500},
7458 ⟨cmr⟩     \guilsinglleft    = {500,300},   \guilsinglright   = {400,400},
7459 ⟨bch⟩     \guillemotleft    = {200,300},   \guillemotright   = {150,400},
7460 ⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright   = {200,300},
7461 ⟨bch⟩     \textbraceleft    = {200,   },   \textbraceright   = {   ,200},
7462 ⟨cmr⟩     \textbraceleft    = {400,100},   \textbraceright   = {200,200},
7463 ⟨bch⟩     \textless         = {100,   },   \textgreater      = {   ,100}
7464 ⟨cmr⟩     \textless         = {300,100},   \textgreater      = {200,100}
7465   }
7466
7467 ⟨/cmr|bch⟩
```

Slanted is very similar to italic.

```
7468 ⟨*cmr⟩
7469 \SetProtrusion
7470   [ name    = cmr-sl,
7471     load    = cmr-it-OT1 ]
7472   { encoding = {OT1,OT4},
7473     family  = cmr,
7474     shape   = sl }
7475   {
7476     L = {  ,50},
7477     f = {  ,-50},
7478     - = {300, },
7479     \textendash = {400,   },  \textemdash = {300,   }
7480   }
7481
7482 \SetProtrusion
7483   [ name    = cmr-sl-T1,
7484     load    = cmr-it-T1 ]
7485   { encoding = {T1,LY1},
7486     family  = cmr,
7487     shape   = sl }
```

```
7488    {
7489      L = {  ,50},
7490      f = { ,-50},
7491      - = {300, },
7492      \textendash = {400, },  \textemdash = {300, }
7493    }
7494
7495 \SetProtrusion
7496    [ name    = cmr-sl-T2A,
7497      load    = cmr-it-T2A ]
7498    { encoding = T2A,
7499      family   = cmr,
7500      shape    = sl  }
7501    {
7502      L = {  ,50},
7503      f = { ,-50},
7504      - = {300, },
7505      \textendash = {400, },  \textemdash = {300, }
7506    }
7507
7508 \SetProtrusion
7509    [ name    = cmr-sl-T5,
7510      load    = cmr-it-T5 ]
7511    { encoding = T5,
7512      family   = cmr,
7513      shape    = sl  }
7514    {
7515      L = {  ,50},
7516      f = { ,-50},
7517      - = {300, },
7518      \textendash = {400, },  \textemdash = {300, }
7519    }
7520
7521 \SetProtrusion
7522    [ name    = lmr-it-T1,
7523      load    = cmr-it-T1 ]
7524    { encoding = {T1,LY1},
7525      family   = lmr,
7526      shape    = {it,sl} }
7527    {
7528      \textquotedblleft  = { ,200},  \textquotedblright = { ,200},
7529      \quotesinglbase    = { ,400},  \quotedblbase      = { ,500}
7530    }
7531
```

Oldstyle numerals are slightly different.

```
7532 \SetProtrusion
7533    [ name = cmr(oldstyle)-it,
7534      load = cmr-it-T1 ]
7535    { encoding = T1,
7536      family   = {hfor,cmor},
7537      shape    = {it,sl}  }
7538    {
7539      1 = {250, 50},
7540      2 = {150,-100},
7541      3 = {100,-50},
7542      4 = {150,150},
7543      6 = {200,   },
7544      7 = {200, 50},
7545      8 = {150,-50},
7546      9 = {100, 50}
7547    }
7548
7549 ⟨/cmr⟩
7550 ⟨*pmn⟩
```

```
7551 \SetProtrusion
7552    [ name     = pmnx-it,
7553      load     = pmnj-it ]
7554    { encoding = OT1,
7555      family   = pmnx,
7556      shape    = {it,sl} }
7557    {
7558      1 = {100,150}
7559    }
7560
7561 \SetProtrusion
7562    [ name     = pmnx-it-T1,
7563      load     = pmnj-it-T1 ]
7564    { encoding = {T1,LY1},
7565      family   = pmnx,
7566      shape    = {it,sl} }
7567    {
7568      1 = {100,150}
7569    }
7570
7571 \SetProtrusion
7572    [ name     = pmnx-it-T2A,
7573      load     = pmnj-it-T2A ]
7574    { encoding = {T2A},
7575      family   = pmnx,
7576      shape    = {it,sl} }
7577    {
7578      1 = {100,150}
7579    }
7580
7581 ⟨/pmn⟩
7582 ⟨*ptm⟩
7583 \SetProtrusion
7584    [ name     = ptm-it-LY1,
7585      load     = ptm-it-T1   ]
7586    { encoding = {LY1},
7587      family   = {ptm,ptmx,ptmj},
7588      shape    = {it,sl}   }
7589    {
7590      _                        = {100,100},
7591      \texttrademark           = {100,100},
7592      \textregistered          = {100,100},
7593      \textcopyright           = {100,100},
7594      \textdegree              = {300,100},
7595      \textminus               = {200,200},
7596      \textellipsis            = {100,200},
7597 %    \texteuro                = {    ,    }, % ?
7598      \textcent                = {100,100},
7599      \textquotesingle         = {500,   },
7600      \textflorin              = {100, 70},
7601      \textdagger              = {150,150},
7602      \textdaggerdbl           = {100,100},
7603      \textbullet              = {150,150},
7604      \textonesuperior         = {150,100},
7605      \texttwosuperior         = {150, 50},
7606      \textthreesuperior       = {150, 50},
7607      \textparagraph           = {100,   },
7608      \textperiodcentered      = {500,300},
7609      \textonequarter          = { 50,   },
7610      \textonehalf             = { 50,   },
7611      \textplusminus           = {100,100},
7612      \textmultiply            = {150,150},
7613      \textdivide              = {150,150}
7614    }
7615
```

```
7616 ⟨/ptm⟩
```

2.8.3 Small caps

Small caps should inherit the values from their big brothers. Since values are relative to character width, we don't need to adjust them any further (but we have to reset some characters).

```
7617 ⟨*!(blg|ugm)⟩
7618 \SetProtrusion
7619 ⟨m-t⟩    [ name      = OT1-sc,
7620 ⟨bch⟩    [ name      = bch-sc,
7621 ⟨cmr⟩    [ name      = cmr-sc-OT1,
7622 ⟨ebg⟩    [ name      = EBGaramond-sc-OT1-Prop,
7623 ⟨pmn⟩    [ name      = pmnj-sc,
7624 ⟨ppl⟩    [ name      = ppl-sc,
7625 ⟨ptm⟩    [ name      = ptm-sc,
7626 ⟨m-t⟩      load      = default ]
7627 ⟨bch⟩      load      = bch-default ]
7628 ⟨cmr⟩      load      = cmr-OT1 ]
7629 ⟨ebg⟩      load      = EBGaramond-OT1-LF ]
7630 ⟨pmn⟩      load      = pmnj-default ]
7631 ⟨ppl⟩      load      = ppl-default ]
7632 ⟨ptm⟩      load      = ptm-default ]
7633 ⟨m-t|bch|ebg|pmn⟩   { encoding = OT1,
7634 ⟨cmr|ppl|ptm⟩   { encoding = {OT1,OT4},
7635 ⟨bch⟩      family    = bch,
7636 ⟨cmr⟩      family    = cmr,
7637 ⟨ebg⟩      family    = {EBGaramond-LF,EBGaramond-OsF},
7638 ⟨pmn⟩      family    = pmnj,
7639 ⟨ppl⟩      family    = {ppl,pplx,pplj},
7640 ⟨ptm⟩      family    = {ptm,ptmx,ptmj},
7641      shape     = sc }
7642    {
7643      a = {50,50},
7644 ⟨cmr|ebg|ppl|ptm⟩    \ae = {50,  },
7645 ⟨bch|pmn⟩       c = {50,  },
7646 ⟨bch|ebg|pmn⟩      d = {  ,50},
7647 ⟨m-t|bch|cmr|ebg|pmn|ptm⟩     f = {  ,50},
7648 ⟨bch|ebg|pmn⟩     g = {50,  },
7649 ⟨m-t|cmr|ebg|pmn|ppl|ptm⟩    j = {50,  },
7650 ⟨bch⟩      j = {100,  },
7651 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩     l = {  ,50},
7652 ⟨ptm⟩       l = {  ,80},
7653 ⟨m-t|bch|cmr|pmn|ppl⟩   013 = {  ,50}, % fl
7654 ⟨ptm⟩     013 = {  ,80}, % fl
7655 ⟨bch|ebg|pmn⟩      o = {50,50},
7656 ⟨ebg|pmn⟩    \oe = {50,  },
7657 ⟨ppl⟩       p = { 0, 0},
7658 ⟨bch|ebg|pmn⟩      q = {50,70},
7659 ⟨ppl⟩      q = { 0,  },
7660 ⟨m-t|cmr|ebg|pmn|ppl|ptm⟩     r = {  , 0},
7661      t = {50,50},
7662 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩    y = {50,50}
7663 ⟨ptm⟩      y = {80,80}
7664    }
7665
7666 ⟨*ebg⟩
7667 \SetProtrusion
7668    [ name      = EBGaramond-sc-OT1-Tab,
7669      load      = EBGaramond-OT1-TOsF ]
7670    { encoding = OT1,
7671      family    = {EBGaramond-TLF,EBGaramond-TOsF},
7672      shape     = sc }
```

```
7673    {
7674      a = {50,50},
7675    \ae = {50,   },
7676      d = {   ,50},
7677      f = {   ,50},
7678      g = {50,   },
7679      j = {50,   },
7680      l = {   ,50},
7681      o = {50,50},
7682    \oe = {50,   },
7683      q = {50,70},
7684      r = {   , 0},
7685      t = {50,50},
7686      y = {50,50}
7687    }
7688
7689 ⟨/ebg⟩
7690 \SetProtrusion
7691 ⟨m-t⟩   [ name    = T1-sc,
7692 ⟨bch⟩   [ name    = bch-sc-T1,
7693 ⟨cmr⟩   [ name    = cmr-sc-T1,
7694 ⟨ebg⟩   [ name    = EBGaramond-sc-T1,
7695 ⟨pmn⟩   [ name    = pmnj-sc-T1,
7696 ⟨ppl⟩   [ name    = ppl-sc-T1,
7697 ⟨ptm⟩   [ name    = ptm-sc-T1,
7698 ⟨m-t⟩    load    = T1-default ]
7699 ⟨bch⟩    load    = bch-T1      ]
7700 ⟨cmr⟩    load    = cmr-T1      ]
7701 ⟨ebg⟩    load    = EBGaramond-T1      ]
7702 ⟨pmn⟩    load    = pmnj-T1     ]
7703 ⟨ppl⟩    load    = ppl-T1      ]
7704 ⟨ptm⟩    load    = ptm-T1      ]
7705 ⟨!ebg⟩   { encoding = {T1,LY1},
7706 ⟨ebg⟩    { encoding = {LY1},
7707 ⟨bch⟩      family   = bch,
7708 ⟨cmr⟩      family   = cmr,
7709 ⟨ebg⟩      family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
7710 ⟨pmn⟩      family   = pmnj,
7711 ⟨ppl⟩      family   = {ppl,pplx,pplj},
7712 ⟨ptm⟩      family   = {ptm,ptmx,ptmj},
7713      shape    = sc }
7714    {
7715      a = {50,50},
7716 ⟨cmr|ebg|ppl|ptm⟩   \ae = {50,   },
7717 ⟨bch|pmn⟩       c = {50,   },
7718 ⟨bch|ebg|pmn⟩      d = {   ,50},
7719 ⟨m-t|bch|cmr|ebg|pmn|ptm⟩      f = {   ,50},
7720 ⟨bch|ebg|pmn⟩      g = {50,   },
7721 ⟨m-t|cmr|ebg|pmn|ppl|ptm⟩      j = {50,   },
7722 ⟨bch⟩      j = {100,   },
7723 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩      l = {   ,50},
7724 ⟨ptm⟩      l = {   ,80},
7725 ⟨m-t|bch|cmr|pmn|ppl⟩   029 = {   ,50}, % fl
7726 ⟨ptm⟩   029 = {   ,80}, % fl
7727 ⟨bch|ebg|pmn⟩       o = {50,50},
7728 ⟨bch|ebg|pmn⟩    \oe = {50,   },
7729 ⟨ppl⟩      p = { 0, 0},
7730 ⟨bch|ebg|pmn⟩      q = {50,70},
7731 ⟨ppl⟩      q = { 0,   },
7732 ⟨m-t|cmr|ebg|pmn|ppl|ptm⟩      r = {   , 0},
7733      t = {50,50},
7734 ⟨m-t|bch|cmr|ebg|pmn|ppl⟩      y = {50,50}
7735 ⟨ptm⟩      y = {80,80}
7736    }
7737
```

```
7738 ⟨/!(blg|ugm)⟩
7739 ⟨*m-t|cmr⟩
7740 \SetProtrusion
7741 ⟨m-t⟩   [ name    = T2A-sc,
7742 ⟨cmr⟩   [ name    = cmr-sc-T2A,
7743 ⟨m-t⟩     load    = T2A-default ]
7744 ⟨cmr⟩     load    = cmr-T2A     ]
7745    { encoding = T2A,
7746 ⟨cmr⟩     family  = cmr,
7747      shape    = sc }
7748    {
7749      \cyra = {50,50},
7750      \cyrg = {  ,50},
7751      \cyrt = {50,50},
7752      \cyry = {  ,50}
7753    }
7754
7755 ⟨/m-t|cmr⟩
7756 ⟨*m-t⟩
7757 \SetProtrusion
7758    [ name    = QX-sc,
7759      load    = QX-default ]
7760    { encoding = QX,
7761      shape    = sc  }
7762    {
7763      a = {50,50},
7764      f = {  ,50},
7765      j = {50,  },
7766      l = {  ,50},
7767    013 = {  ,50}, % fl
7768      r = {  , 0},
7769      t = {50,50},
7770      y = {50,50}
7771    }
7772
7773 ⟨/m-t⟩
7774 ⟨*cmr|bch⟩
7775 \SetProtrusion
7776 ⟨bch⟩   [ name     = bch-sc-T5,
7777 ⟨bch⟩     load     = bch-T5 ]
7778 ⟨cmr⟩   [ name     = cmr-sc-T5,
7779 ⟨cmr⟩     load     = cmr-T5 ]
7780    { encoding = T5,
7781 ⟨bch⟩     family   = bch,
7782 ⟨cmr⟩     family   = cmr,
7783      shape    = sc }
7784    {
7785      a = {50,50},
7786 ⟨bch⟩     c = {50,  },
7787 ⟨bch⟩     d = {  ,50},
7788      f = {  ,50},
7789 ⟨bch⟩     g = {50,  },
7790 ⟨bch⟩     j = {100,  },
7791 ⟨cmr⟩     j = {50,  },
7792      l = {  ,50},
7793 ⟨bch⟩     o = {50,50},
7794 ⟨bch⟩     q = { 0,  },
7795 ⟨cmr⟩     r = {  , 0},
7796      t = {50,50},
7797      y = {50,50}
7798    }
7799
7800 ⟨/cmr|bch⟩
7801 ⟨*ebg⟩
7802 \SetProtrusion
```

```
7803    [ name    = EBGaramond-sc-T1-Prop,
7804      load    = EBGaramond-T1-LF ]
7805    { encoding = T1,
7806      family  = {EBGaramond-LF,EBGaramond-OsF},
7807      shape   = sc }
7808    {
7809      a = {50,50},
7810    \ae = {50,  },
7811      d = {  ,50},
7812      f = {  ,50},
7813      g = {50,  },
7814      j = {50,  },
7815      l = {  ,50},
7816      o = {50,50},
7817    \oe = {50,  },
7818      q = {50,70},
7819      r = {  , 0},
7820      t = {50,50},
7821      y = {50,50}
7822    }
7823
7824 \SetProtrusion
7825    [ name    = EBGaramond-sc-T1-Tab,
7826      load    = EBGaramond-T1-TOsF ]
7827    { encoding = T1,
7828      family  = {EBGaramond-TLF,EBGaramond-TOsF},
7829      shape   = sc }
7830    {
7831      a = {50,50},
7832    \ae = {50,  },
7833      d = {  ,50},
7834      f = {  ,50},
7835      g = {50,  },
7836      j = {50,  },
7837      l = {  ,50},
7838      o = {50,50},
7839    \oe = {50,  },
7840      q = {50,70},
7841      r = {  , 0},
7842      t = {50,50},
7843      y = {50,50}
7844    }
7845
7846 ⟨/ebg⟩
7847 ⟨*pmn⟩
7848 \SetProtrusion
7849    [ name    = pmnx-sc,
7850      load    = pmnj-sc ]
7851    { encoding = OT1,
7852      family  = pmnx,
7853      shape   = sc }
7854    {
7855      1 = {230,180}
7856    }
7857
7858 \SetProtrusion
7859    [ name    = pmnx-sc-T1,
7860      load    = pmnj-sc-T1 ]
7861    { encoding = {T1,LY1},
7862      family  = pmnx,
7863      shape   = sc }
7864    {
7865      1 = {230,180}
7866    }
7867
```

2.8.4 Italic small caps

Minion provides real small caps in italics. The slantsc package calls them scit,
Philipp Lehman's fontinstallationguide suggests si.

```
7868 \SetProtrusion
7869   [ name     = pmnj-scit,
7870     load     = pmnj-it   ]
7871   { encoding = OT1,
7872     family   = pmnj,
7873     shape    = {scit,si} }
7874   {
7875     a = {50,  },
7876   \ae = {  ,-50},
7877     b = {20,-50},
7878     c = {50,-50},
7879     d = {20, 0},
7880     e = {20,-50},
7881     f = {10, 0},
7882   012 = {10,-50}, % fi
7883   013 = {10,-50}, % fl
7884   014 = {10,-50}, % ffi
7885   015 = {10,-50}, % ffl
7886     g = {50,-50},
7887     i = {20,-50},
7888     j = {20, 0},
7889     k = {20,  },
7890     l = {20,50},
7891     m = {  ,-30},
7892     n = {  ,-30},
7893     o = {50,  },
7894   \oe = {50,-50},
7895     p = {20,-50},
7896     q = {50,  },
7897     r = {20, 0},
7898     s = {20,-30},
7899     t = {70,  },
7900     u = {50,-50},
7901     v = {100,  },
7902     w = {100,  },
7903     y = {50,  },
7904     z = {  ,-50}
7905   }
7906
7907 \SetProtrusion
7908   [ name     = pmnj-scit-T1,
7909     load     = pmnj-it-T1   ]
7910   { encoding = {T1,LY1},
7911     family   = pmnj,
7912     shape    = {scit,si}    }
7913   {
7914     a = {50,  },
7915   \ae = {  ,-50},
7916     b = {20,-50},
7917     c = {50,-50},
7918     d = {20, 0},
7919     e = {20,-50},
7920     f = {10, 0},
7921   028 = {10,-50}, % fi
7922   029 = {10,-50}, % fl
7923   030 = {10,-50}, % ffi
7924   031 = {10,-50}, % ffl
7925     g = {50,-50},
7926     i = {20,-50},
7927   188 = {20, 0},  % ij
7928     j = {20, 0},
```

```
7929        k = {20,  },
7930        l = {20,50},
7931        m = {  ,-30},
7932        n = {  ,-30},
7933        o = {50,  },
7934      \oe = {50,-50},
7935        p = {20,-50},
7936        q = {50,  },
7937        r = {20, 0},
7938        s = {20,-30},
7939        t = {70,  },
7940        u = {50,-50},
7941        v = {100,  },
7942        w = {100,  },
7943        y = {50,  },
7944        z = {  ,-50}
7945      }
7946
7947 \SetProtrusion
7948    [ name     = pmnx-scit,
7949      load     = pmnj-scit ]
7950    { encoding = OT1,
7951      family   = pmnx,
7952      shape    = {scit,si} }
7953    {
7954      1 = {100,150}
7955    }
7956
7957 \SetProtrusion
7958    [ name     = pmnx-scit-T1,
7959      load     = pmnj-scit-T1 ]
7960    { encoding = {T1,LY1},
7961      family   = pmnx,
7962      shape    = {scit,si}     }
7963    {
7964      1 = {100,150}
7965    }
7966
7967 ⟨/pmn⟩
7968 ⟨*ebg⟩
```

For small caps italics, we copy the definitions from the small caps settings, except that we first load the italics settings.

```
7969 \SetProtrusion
7970    [ name     = EBGaramond-scit-OT1-Prop,
7971      load     = EBGaramond-it-OT1-LF ]
7972    { encoding = OT1,
7973      family   = {EBGaramond-LF,EBGaramond-OsF},
7974      shape    = scit }
7975    {
7976      a = {50,50},
7977    \ae = {50,  },
7978      d = {  ,50},
7979      f = {  ,50},
7980      g = {50,  },
7981      j = {50,  },
7982      l = {  ,50},
7983      o = {50,50},
7984    \oe = {50,  },
7985      q = {50,70},
7986      r = {  , 0},
7987      t = {50,50},
7988      y = {50,50}
7989    }
7990
```

```
7991 \SetProtrusion
7992    [ name     = EBGaramond-scit-OT1-Tab,
7993      load     = EBGaramond-it-OT1-TOsF ]
7994    { encoding = OT1,
7995      family   = {EBGaramond-TLF,EBGaramond-TOsF},
7996      shape    = scit }
7997    {
7998      a = {50,50},
7999    \ae = {50,  },
8000      d = {  ,50},
8001      f = {  ,50},
8002      g = {50,  },
8003      j = {50,  },
8004      l = {  ,50},
8005      o = {50,50},
8006    \oe = {50,  },
8007      q = {50,70},
8008      r = {  , 0},
8009      t = {50,50},
8010      y = {50,50}
8011    }
8012
8013 \SetProtrusion
8014    [ name     = EBGaramond-scit-T1-Prop,
8015      load     = EBGaramond-it-T1-LF ]
8016    { encoding = T1,
8017      family   = {EBGaramond-LF,EBGaramond-OsF},
8018      shape    = scit }
8019    {
8020      a = {50,50},
8021    \ae = {50,  },
8022      d = {  ,50},
8023      f = {  ,50},
8024      g = {50,  },
8025      j = {50,  },
8026      l = {  ,50},
8027      o = {50,50},
8028    \oe = {50,  },
8029      q = {50,70},
8030      r = {  , 0},
8031      t = {50,50},
8032      y = {50,50}
8033    }
8034
8035 \SetProtrusion
8036    [ name     = EBGaramond-scit-T1-Tab,
8037      load     = EBGaramond-it-T1-TOsF ]
8038    { encoding = T1,
8039      family   = {EBGaramond-TLF,EBGaramond-TOsF},
8040      shape    = scit }
8041    {
8042      a = {50,50},
8043    \ae = {50,  },
8044      d = {  ,50},
8045      f = {  ,50},
8046      g = {50,  },
8047      j = {50,  },
8048      l = {  ,50},
8049      o = {50,50},
8050    \oe = {50,  },
8051      q = {50,70},
8052      r = {  , 0},
8053      t = {50,50},
8054      y = {50,50}
8055    }
```

```
8056
8057 ⟨/ebg⟩
```

2.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```
8058 \SetProtrusion
8059 ⟨m-t⟩   [ name     = textcomp ]
8060 ⟨bch⟩   [ name     = bch-textcomp ]
8061 ⟨blg⟩   [ name     = blg-textcomp ]
8062 ⟨cmr⟩   [ name     = cmr-textcomp ]
8063 ⟨ebg⟩   [ name     = EBGaramond-textcomp ]
8064 ⟨pmn⟩   [ name     = pmn-textcomp ]
8065 ⟨ppl⟩   [ name     = ppl-textcomp ]
8066 ⟨ptm⟩   [ name     = ptm-textcomp ]
8067 ⟨ugm⟩   [ name     = ugm-textcomp ]
8068 ⟨m-t⟩   { encoding = TS1      }
8069 ⟨!m-t⟩  { encoding = TS1,
8070 ⟨bch⟩     family   = bch }
8071 ⟨blg⟩     family   = blg }
8072 ⟨cmr⟩     family   = cmr }
8073 ⟨ebg⟩     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF} }
8074 ⟨pmn⟩     family   = {pmnx,pmnj} }
8075 ⟨ppl⟩     family   = {ppl,pplx,pplj} }
8076 ⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
8077 ⟨ugm⟩     family   = ugm }
8078    {
8079 ⟨blg⟩     \textquotestraightbase    = {400,500},
8080 ⟨cmr⟩     \textquotestraightbase    = {300,300},
8081 ⟨ebg|pmn⟩    \textquotestraightbase    = {400,400},
8082 ⟨blg⟩     \textquotestraightdblbase = {300,400},
8083 ⟨cmr|pmn⟩    \textquotestraightdblbase = {300,300},
8084 ⟨ebg⟩     \textquotestraightdblbase = {400,400},
8085 ⟨bch|cmr|ebg|pmn|ugm⟩    \texttwelveudash         = {200,200},
8086 ⟨bch|cmr|ebg|pmn⟩    \textthreequartersemdash = {150,150},
8087 ⟨ugm⟩     \textthreequartersemdash = {200,200},
8088 ⟨blg⟩     \textquotesingle          = {500,600},
8089 ⟨cmr|pmn⟩    \textquotesingle          = {300,400},
8090 ⟨ebg⟩     \textquotesingle          = {400,500},
8091 ⟨ptm⟩     \textquotesingle          = {500,500},
8092 ⟨ugm⟩     \textquotesingle          = {300,500},
8093 ⟨bch|cmr|pmn⟩    \textasteriskcentered     = {200,300},
8094 ⟨blg⟩     \textasteriskcentered     = {150,200},
8095 ⟨ebg⟩     \textasteriskcentered     = {300,300},
8096 ⟨ugm⟩     \textasteriskcentered     = {100,200},
8097 ⟨pmn⟩     \textfractionsolidus      = {-200,-200},
8098 ⟨cmr⟩     \textoneoldstyle          = {100,100},
8099 ⟨pmn⟩     \textoneoldstyle          = {  , 50},
8100 ⟨cmr⟩     \textthreeoldstyle        = {  , 50},
8101 ⟨ebg|pmn⟩    \textthreeoldstyle        = { 50,  },
8102 ⟨cmr⟩     \textfouroldstyle         = { 50, 50},
8103 ⟨ebg|pmn⟩    \textfouroldstyle         = { 50,  },
8104 ⟨cmr|ebg|pmn⟩    \textsevenoldstyle        = { 50, 80},
8105 ⟨cmr⟩     \textlangle               = {400,  },
8106 ⟨cmr⟩     \textrangle               = {  ,400},
8107 ⟨m-t|bch|pmn|ptm⟩    \textminus               = {200,200},
8108 ⟨cmr|ebg|ppl⟩    \textminus               = {300,300},
8109 ⟨blg|ugm⟩    \textminus               = {250,300},
8110 ⟨bch|ebg|pmn⟩    \textlbrackdbl            = {100,  },
8111 ⟨blg⟩     \textlbrackdbl            = {200,  },
8112 ⟨bch|ebg|pmn⟩    \textrbrackdbl            = {  ,100},
8113 ⟨blg⟩     \textrbrackdbl            = {  ,200},
8114 ⟨pmn⟩     \textasciigrave           = {200,500},
```

```
8115 ⟨bch|blg|cmr|ebg|pmn⟩       \texttildelow                = {200,250},
8116 ⟨pmn⟩       \textasciibreve          = {300,400},
8117 ⟨pmn⟩       \textasciicaron          = {300,400},
8118 ⟨pmn⟩       \textacutedbl            = {200,300},
8119 ⟨pmn⟩       \textgravedbl            = {150,300},
8120 ⟨bch|pmn|ugm⟩       \textdagger              = { 80, 80},
8121 ⟨blg⟩       \textdagger              = {200,200},
8122 ⟨cmr|ebg⟩       \textdagger              = {100,100},
8123 ⟨ptm⟩       \textdagger              = {150,150},
8124 ⟨blg⟩       \textdaggerdbl           = {150,150},
8125 ⟨cmr|ebg|pmn⟩       \textdaggerdbl           = { 80, 80},
8126 ⟨ptm⟩       \textdaggerdbl           = {100,100},
8127 ⟨bch⟩       \textbardbl              = {100,100},
8128 ⟨blg|ugm⟩       \textbardbl              = {150,150},
8129 ⟨bch⟩       \textbullet              = {200,200},
8130 ⟨blg⟩       \textbullet              = {400,500},
8131 ⟨cmr|ebg|pmn⟩       \textbullet              = {    ,100},
8132 ⟨ptm⟩       \textbullet              = {150,150},
8133 ⟨ugm⟩       \textbullet              = { 50,100},
8134 ⟨bch|cmr|pmn⟩       \textcelsius             = { 50,    },
8135 ⟨ebg⟩       \textcelsius             = { 80,    },
8136 ⟨bch⟩       \textflorin              = { 50, 50},
8137 ⟨blg⟩       \textflorin              = {100,100},
8138 ⟨ebg|ugm⟩       \textflorin              = {    ,100},
8139 ⟨pmn⟩       \textflorin              = { 50,100},
8140 ⟨ptm⟩       \textflorin              = { 50, 70},
8141 ⟨cmr⟩       \textcolonmonetary       = {    , 50},
8142 ⟨ebg|pmn⟩       \textcolonmonetary       = { 50,    },
8143 ⟨pmn⟩       \textinterrobang         = {    ,100},
8144 ⟨pmn⟩       \textinterrobangdown     = {100,    },
8145 ⟨m-t|ebg|ptm⟩       \texttrademark           = {100,100},
8146 ⟨bch⟩       \texttrademark           = {150,150},
8147 ⟨blg|cmr|ppl⟩       \texttrademark           = {200,200},
8148 ⟨pmn⟩       \texttrademark           = { 50, 50},
8149 ⟨ugm⟩       \texttrademark           = {100,150},
8150 ⟨bch|ugm⟩       \textcent                = { 50,    },
8151 ⟨ptm⟩       \textcent                = {100,100},
8152 ⟨bch⟩       \textsterling            = { 50,    },
8153 ⟨ugm⟩       \textsterling            = {    , 50},
8154 ⟨bch⟩       \textbrokenbar           = {200,200},
8155 ⟨blg⟩       \textbrokenbar           = {250,250},
8156 ⟨ugm⟩       \textbrokenbar           = {200,300},
8157 ⟨pmn⟩       \textasciidieresis       = {300,400},
8158 ⟨m-t|bch|cmr|ebg|ptm|ugm⟩       \textcopyright           = {100,100},
8159 ⟨pmn⟩       \textcopyright           = {100,150},
8160 ⟨ppl⟩       \textcopyright           = {200,200},
8161 ⟨bch|cmr|ugm⟩       \textordfeminine         = {100,200},
8162 ⟨ebg|pmn⟩       \textordfeminine         = {200,200},
8163 ⟨bch|cmr|ebg|pmn|ugm⟩       \textlnot                = {200,    },
8164 ⟨blg⟩       \textlnot                = {200,100},
8165 ⟨m-t|bch|cmr|ebg|ptm|ugm⟩       \textregistered          = {100,100},
8166 ⟨pmn⟩       \textregistered          = { 50,150},
8167 ⟨ppl⟩       \textregistered          = {200,200},
8168 ⟨pmn⟩       \textasciimacron         = {150,200},
8169 ⟨m-t|ppl|ptm⟩       \textdegree              = {300,300},
8170 ⟨bch⟩       \textdegree              = {150,200},
8171 ⟨blg|ugm⟩       \textdegree              = {200,200},
8172 ⟨cmr|ebg⟩       \textdegree              = {400,400},
8173 ⟨pmn⟩       \textdegree              = {150,400},
8174 ⟨bch|cmr|ebg|pmn|ugm⟩       \textpm                  = {150,200},
8175 ⟨blg⟩       \textpm                  = {100,100},
8176 ⟨ptm⟩       \textpm                  = { 50, 80},
8177 ⟨bch|blg|ugm⟩       \texttwosuperior         = {100,200},
8178 ⟨cmr⟩       \texttwosuperior         = { 50,100},
8179 ⟨ebg|pmn⟩       \texttwosuperior         = {200,200},
```

```
8180 ⟨ptm⟩     \texttwosuperior           = { 50, 50},
8181 ⟨bch|blg|ugm⟩    \textthreesuperior        = {100,200},
8182 ⟨cmr⟩     \textthreesuperior         = { 50,100},
8183 ⟨ebg|pmn⟩    \textthreesuperior        = {200,200},
8184 ⟨ptm⟩     \textthreesuperior         = { 50, 50},
8185 ⟨pmn⟩     \textasciiacute            = {300,400},
8186 ⟨bch|ugm⟩    \textmu                   = {    ,100},
8187 ⟨bch|ebg|pmn⟩    \textparagraph        = {    ,100},
8188 ⟨bch|cmr|ebg|pmn⟩    \textperiodcentered    = {300,400},
8189 ⟨blg⟩     \textperiodcentered        = {400,500},
8190 ⟨ptm⟩     \textperiodcentered        = {300,300},
8191 ⟨ugm⟩     \textperiodcentered        = {200,500},
8192 ⟨bch|blg|ugm⟩    \textonesuperior          = {200,300},
8193 ⟨cmr|ebg|pmn⟩    \textonesuperior          = {200,200},
8194 ⟨ptm⟩     \textonesuperior           = {100,100},
8195 ⟨bch|ebg|pmn|ugm⟩    \textordmasculine      = {200,200},
8196 ⟨blg|cmr⟩    \textordmasculine         = {100,200},
8197 ⟨bch|cmr|pmn⟩    \texteuro                 = {100,    },
8198 ⟨ebg⟩     \texteuro                  = { 50,100},
8199 ⟨bch⟩     \texttimes                 = {200,200},
8200 ⟨blg|ptm⟩    \texttimes                = {100,100},
8201 ⟨cmr⟩     \texttimes                 = {150,250},
8202 ⟨ebg⟩     \texttimes                 = {100,150},
8203 ⟨pmn⟩     \texttimes                 = { 70,100},
8204 ⟨ugm⟩     \texttimes                 = {200,300},
8205 ⟨bch|ebg|pmn⟩    \textdiv                  = {150,200}
8206 ⟨blg⟩     \textdiv                   = {100,100}
8207 ⟨cmr⟩     \textdiv                   = {150,250}
8208 ⟨ptm⟩     \textdiv                   = { 50,100},
8209 ⟨ugm⟩     \textdiv                   = {200,300},
8210 ⟨ptm⟩     \textperthousand           = {    ,50}
8211 ⟨ugm⟩     \textsection               = {    ,100},
8212 ⟨ugm⟩     \textonehalf               = { 50,100},
8213 ⟨ugm⟩     \textonequarter            = { 50,100},
8214 ⟨ugm⟩     \textthreequarters         = { 50,100},
8215 ⟨ugm⟩     \textsurd                  = {    ,100}
```

Remaining slots in the source file.

```
8216     }
8217
8218 ⟨*cmr|ebg|pmn|ugm⟩
8219 \SetProtrusion
8220 ⟨cmr⟩    [ name     = cmr-textcomp-it ]
8221 ⟨ebg⟩    [ name     = EBGaramond-textcomp-it ]
8222 ⟨pmn⟩    [ name     = pmn-textcomp-it ]
8223 ⟨ugm⟩    [ name     = ugm-textcomp-it ]
8224     { encoding = TS1,
8225 ⟨cmr⟩       family    = cmr,
8226 ⟨ebg⟩       family    = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
8227 ⟨pmn⟩       family    = {pmnx,pmnj},
8228 ⟨ugm⟩       family    = ugm,
8229 ⟨cmr|pmn⟩      shape    = {it,sl} }
8230 ⟨ebg|ugm⟩      shape    = it }
8231     {
8232 ⟨cmr⟩      \textquotestraightbase    = {300,600},
8233 ⟨ebg|pmn⟩    \textquotestraightbase   = {400,400},
8234 ⟨cmr⟩      \textquotestraightdblbase = {300,600},
8235 ⟨ebg⟩      \textquotestraightdblbase = {300,400},
8236 ⟨pmn⟩      \textquotestraightdblbase = {300,300},
8237      \texttwelveudash           = {200,200},
8238 ⟨cmr|ebg|pmn⟩    \textthreequartersemdash  = {150,150},
8239 ⟨ugm⟩      \textthreequartersemdash  = {200,200},
8240 ⟨cmr⟩      \textquotesingle          = {600,300},
8241 ⟨ebg⟩      \textquotesingle          = {800,100},
8242 ⟨pmn⟩      \textquotesingle          = {300,200},
```

```
8243 ⟨ugm⟩    \textquotesingle          = {500,500},
8244 ⟨cmr⟩    \textasteriskcentered     = {300,200},
8245 ⟨ebg⟩    \textasteriskcentered     = {500,100},
8246 ⟨pmn⟩    \textasteriskcentered     = {200,300},
8247 ⟨ugm⟩    \textasteriskcentered     = {300,150},
8248 ⟨pmn⟩    \textfractionsolidus      = {-200,-200},
8249 ⟨cmr⟩    \textoneoldstyle          = {100, 50},
8250 ⟨ebg⟩    \textoneoldstyle          = {100,   },
8251 ⟨pmn⟩    \textoneoldstyle          = { 50,   },
8252 ⟨ebg⟩    \texttwooldstyle          = { 50,   },
8253 ⟨pmn⟩    \texttwooldstyle          = {-50,   },
8254 ⟨cmr⟩    \textthreeoldstyle        = {100, 50},
8255 ⟨pmn⟩    \textthreeoldstyle        = {-100,  },
8256 ⟨cmr⟩    \textfouroldstyle         = { 50, 50},
8257 ⟨ebg⟩    \textfouroldstyle         = { 50,100},
8258 ⟨cmr⟩    \textsevenoldstyle        = { 50, 80},
8259 ⟨ebg⟩    \textsevenoldstyle        = { 50,   },
8260 ⟨pmn⟩    \textsevenoldstyle        = { 20,   },
8261 ⟨cmr⟩    \textlangle               = {400,   },
8262 ⟨cmr⟩    \textrangle               = {   ,400},
8263 ⟨cmr|ebg⟩     \textminus               = {300,300},
8264 ⟨pmn⟩    \textminus                = {200,200},
8265 ⟨ugm⟩    \textminus                = {250,300},
8266 ⟨ebg|pmn⟩     \textlbrackdbl          = {100,   },
8267 ⟨ebg|pmn⟩     \textrbrackdbl          = {   ,100},
8268 ⟨pmn⟩    \textasciigrave           = {300,300},
8269 ⟨cmr|ebg|pmn⟩     \texttildelow          = {200,250},
8270 ⟨pmn⟩    \textasciibreve           = {300,300},
8271 ⟨pmn⟩    \textasciicaron           = {300,300},
8272 ⟨pmn⟩    \textacutedbl             = {200,300},
8273 ⟨pmn⟩    \textgravedbl             = {150,300},
8274 ⟨cmr⟩    \textdagger               = {100,100},
8275 ⟨ebg⟩    \textdagger               = {200,100},
8276 ⟨pmn⟩    \textdagger               = { 80, 50},
8277 ⟨ugm⟩    \textdagger               = { 80, 80},
8278 ⟨cmr|ebg⟩     \textdaggerdbl          = { 80, 80},
8279 ⟨pmn⟩    \textdaggerdbl            = { 80, 50},
8280 ⟨ugm⟩    \textbardbl               = {150,150},
8281 ⟨cmr⟩    \textbullet               = {200,100},
8282 ⟨ebg⟩    \textbullet               = {300,   },
8283 ⟨pmn⟩    \textbullet               = { 30, 70},
8284 ⟨ugm⟩    \textbullet               = { 50,100},
8285 ⟨cmr⟩    \textcelsius              = {100,   },
8286 ⟨ebg⟩    \textcelsius              = {200,   },
8287 ⟨pmn⟩    \textcelsius              = { 50,-50},
8288 ⟨ebg⟩    \textflorin               = {100,   },
8289 ⟨pmn⟩    \textflorin               = { 50,100},
8290 ⟨ugm⟩    \textflorin               = {   ,100},
8291 ⟨cmr⟩    \textcolonmonetary        = {150,   },
8292 ⟨ebg⟩    \textcolonmonetary        = {100,   },
8293 ⟨pmn⟩    \textcolonmonetary        = { 50,-50},
8294 ⟨cmr|ebg⟩     \texttrademark          = {200,   },
8295 ⟨pmn⟩    \texttrademark            = { 50,100},
8296 ⟨ugm⟩    \texttrademark            = {150, 50},
8297 ⟨ugm⟩    \textcent                 = { 50,   },
8298 ⟨ugm⟩    \textsterling             = {   , 50},
8299 ⟨ugm⟩    \textbrokenbar            = {200,300},
8300 ⟨pmn⟩    \textasciidieresis        = {300,200},
8301 ⟨cmr⟩    \textcopyright            = {100,   },
8302 ⟨ebg⟩    \textcopyright            = {200,100},
8303 ⟨pmn⟩    \textcopyright            = {100,150},
8304 ⟨ugm⟩    \textcopyright            = {300,   },
8305 ⟨cmr⟩    \textordfeminine          = {100,100},
8306 ⟨pmn⟩    \textordfeminine          = {200,200},
8307 ⟨ugm⟩    \textordfeminine          = {100,200},
```

```
8308 ⟨cmr|ebg⟩      \textlnot                    = {300,   },
8309 ⟨pmn|ugm⟩      \textlnot                    = {200,   },
8310 ⟨cmr⟩      \textregistered          = {100,   },
8311 ⟨ebg⟩      \textregistered          = {200,100},
8312 ⟨pmn⟩      \textregistered          = { 50,150},
8313 ⟨ugm⟩      \textregistered          = {300,   },
8314 ⟨pmn⟩      \textasciimacron         = {150,200},
8315 ⟨cmr|ebg⟩      \textdegree              = {500,100},
8316 ⟨pmn⟩      \textdegree              = {150,150},
8317 ⟨ugm⟩      \textdegree              = {300,200},
8318 ⟨cmr⟩      \textpm                  = {150,100},
8319 ⟨ebg⟩      \textpm                  = {200,150},
8320 ⟨pmn|ugm⟩      \textpm                  = {150,200},
8321 ⟨cmr⟩      \textonesuperior         = {400,   },
8322 ⟨ebg⟩      \textonesuperior         = {300,100},
8323 ⟨pmn⟩      \textonesuperior         = {200,100},
8324 ⟨ugm⟩      \textonesuperior         = {300,300},
8325 ⟨cmr⟩      \texttwosuperior         = {400,   },
8326 ⟨ebg⟩      \texttwosuperior         = {300,   },
8327 ⟨pmn⟩      \texttwosuperior         = {200,100},
8328 ⟨ugm⟩      \texttwosuperior         = {300,200},
8329 ⟨cmr⟩      \textthreesuperior       = {400,   },
8330 ⟨ebg⟩      \textthreesuperior       = {300,   },
8331 ⟨pmn⟩      \textthreesuperior       = {200,100},
8332 ⟨ugm⟩      \textthreesuperior       = {300,200},
8333 ⟨ugm⟩      \textmu                  = {   ,100},
8334 ⟨pmn⟩      \textasciiacute          = {300,200},
8335 ⟨cmr⟩      \textparagraph           = {200,   },
8336 ⟨pmn⟩      \textparagraph           = {   ,100},
8337 ⟨cmr⟩      \textperiodcentered      = {500,500},
8338 ⟨ebg|pmn|ugm⟩      \textperiodcentered      = {300,400},
8339 ⟨cmr⟩      \textordmasculine        = {100,100},
8340 ⟨pmn⟩      \textordmasculine        = {200,200},
8341 ⟨ugm⟩      \textordmasculine        = {300,200},
8342 ⟨cmr⟩      \texteuro                = {200,   },
8343 ⟨ebg⟩      \texteuro                = {100,   },
8344 ⟨pmn⟩      \texteuro                = {100,-50},
8345 ⟨cmr⟩      \texttimes               = {200,200},
8346 ⟨ebg⟩      \texttimes               = {200,100},
8347 ⟨pmn⟩      \texttimes               = { 70,100},
8348 ⟨ugm⟩      \texttimes               = {200,300},
8349 ⟨cmr|ebg⟩      \textdiv                 = {200,200}
8350 ⟨pmn⟩      \textdiv                 = {150,200}
8351 ⟨ugm⟩      \textdiv                 = {200,300},
8352 ⟨ugm⟩      \textsection             = {   ,200},
8353 ⟨ugm⟩      \textonehalf             = { 50,100},
8354 ⟨ugm⟩      \textonequarter          = { 50,100},
8355 ⟨ugm⟩      \textthreequarters       = { 50,100},
8356 ⟨ugm⟩      \textsurd                = {   ,100}
8357    }
8358
8359 ⟨/cmr|ebg|pmn|ugm⟩
```

2.8.6   Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from fontmath.ltx. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font 'operators' (also used for the \mathrm and \mathbf alphabets) is OT1/cmr, which we've already set up above. It's declared as:

```
\DeclareSymbolFont{operators}  {OT1}{cmr}{m} {n}
\SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
```

\mathit (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for \mathsf and \mathtt.

Math font 'letters' (also used as \mathnormal) is declared as:

```
\DeclareSymbolFont{letters}     {OML}{cmm}{m}{it}
\SetSymbolFont{letters}  {bold}{OML}{cmm}{b}{it}
```

```
8360 ⟨*cmr⟩
8361 \SetProtrusion
8362   [ name     = cmr-math-letters ]
8363   { encoding = OML,
8364     family   = cmm,
8365     series   = {m,b},
8366     shape    = it    }
8367   {
8368       A = {100, 50}, % \mathnormal
8369       B = { 50,   },
8370       C = { 50,   },
8371       D = { 50, 50},
8372       E = { 50,   },
8373       F = {100, 50},
8374       G = { 50, 50},
8375       H = { 50, 50},
8376       I = { 50, 50},
8377       J = {150, 50},
8378       K = { 50,100},
8379       L = { 50, 50},
8380       M = { 50,   },
8381       N = { 50,   },
8382       O = { 50,   },
8383       P = { 50,   },
8384       Q = { 50, 50},
8385       R = { 50,   },
8386       S = { 50,   },
8387       T = { 50,100},
8388       U = { 50, 50},
8389       V = {100,100},
8390       W = { 50,100},
8391       X = { 50,100},
8392       Y = {100,100},
8393       f = {100,100},
8394       h = {   ,100},
8395       i = {   , 50},
8396       j = {   , 50},
8397       k = {   , 50},
8398       r = {   , 50},
8399       v = {   , 50},
8400       w = {   , 50},
8401       x = {   , 50},
8402     "0B = { 50,100}, % \alpha
8403     "0C = { 50, 50}, % \beta
8404     "0D = {200,150}, % \gamma
8405     "0E = { 50, 50}, % \delta
8406     "0F = { 50, 50}, % \epsilon
8407     "10 = { 50,150}, % \zeta
8408     "12 = { 50,   }, % \theta
8409     "13 = {   ,100}, % \iota
8410     "14 = {   ,100}, % \kappa
8411     "15 = {100, 50}, % \lambda
8412     "16 = {   , 50}, % \mu
8413     "17 = {   , 50}, % \nu
8414     "18 = {   , 50}, % \xi
8415     "19 = { 50,100}, % \pi
8416     "1A = { 50, 50}, % \rho
8417     "1B = {   ,150}, % \sigma
```

```
8418      "1C = {  50,150}, % \tau
8419      "1D = {  50, 50}, % \upsilon
8420      "1F = {  50,100}, % \chi
8421      "20 = {  50, 50}, % \psi
8422      "21 = {    , 50}, % \omega
8423      "22 = {    , 50}, % \varepsilon
8424      "23 = {    , 50}, % \vartheta
8425      "24 = {    , 50}, % \varpi
8426      "25 = {100,    }, % \varrho
8427      "26 = {100,100}, % \varsigma
8428      "27 = {  50, 50}, % \varphi
8429      "28 = {100,100}, % \leftharpoonup
8430      "29 = {100,100}, % \leftharpoondown
8431      "2A = {100,100}, % \rightharpoonup
8432      "2B = {100,100}, % \rightharpoondown
8433      "2C = {300,200}, % \lhook
8434      "2D = {200,300}, % \rhook
8435      "2E = {    ,100}, % \triangleright
8436      "2F = {100,    }, % \triangleleft
8437      "3A = {    ,500}, % ., \ldotp
8438      "3B = {    ,500}, % ,
8439      "3C = {200,100}, % <
8440      "3D = {300,400}, % /
8441      "3E = {100,200}, % >
8442      "3F = {200,200}, % \star
8443      "5B = {    ,100}, % \flat
8444      "5E = {200,200}, % \smile
8445      "5F = {200,200}, % \frown
8446      "7C = {100,    }, % \jmath
8447      "7D = {    ,100}  % \wp
```

Remaining slots in the source file.

```
8448      }
8449
```

Math font 'symbols' (also used for the \mathcal alphabet) is declared as:

```
\DeclareSymbolFont{symbols}     {OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols}   {bold}{OMS}{cmsy}{b}{n}
```

```
8450  \SetProtrusion
8451    [ name      = cmr-math-symbols ]
8452    { encoding = OMS,
8453      family   = cmsy,
8454      series   = {m,b},
8455      shape    = n  }
8456    {
8457      A = {150, 50}, % \mathcal
8458      C = {    ,100},
8459      D = {    , 50},
8460      F = {  50,150},
8461      I = {    ,100},
8462      J = {100,150},
8463      K = {    ,100},
8464      L = {100,    },
8465      M = {  50, 50},
8466      N = {  50,100},
8467      P = {    , 50},
8468      Q = {  50,    },
8469      R = {    , 50},
8470      T = {  50,150},
8471      V = {  50, 50},
8472      W = {    , 50},
8473      X = {100,100},
8474      Y = {100,    },
8475      Z = {100,150},
```

```
8476    "00 = {300,300}, % -
8477    "01 = {    ,700}, % \cdot, \cdotp
8478    "02 = {150,250}, % \times
8479    "03 = {150,250}, % *, \ast
8480    "04 = {200,300}, % \div
8481    "05 = {150,250}, % \diamond
8482    "06 = {200,200}, % \pm
8483    "07 = {200,200}, % \mp
8484    "08 = {100,100}, % \oplus
8485    "09 = {100,100}, % \ominus
8486    "0A = {100,100}, % \otimes
8487    "0B = {100,100}, % \oslash
8488    "0C = {100,100}, % \odot
8489    "0D = {100,100}, % \bigcirc
8490    "0E = {100,100}, % \circ
8491    "0F = {100,100}, % \bullet
8492    "10 = {100,100}, % \asymp
8493    "11 = {100,100}, % \equiv
8494    "12 = {200,100}, % \subseteq
8495    "13 = {100,200}, % \supseteq
8496    "14 = {200,100}, % \leq
8497    "15 = {100,200}, % \geq
8498    "16 = {200,100}, % \preceq
8499    "17 = {100,200}, % \succeq
8500    "18 = {200,200}, % \sim
8501    "19 = {150,150}, % \approx
8502    "1A = {200,100}, % \subset
8503    "1B = {100,200}, % \supset
8504    "1C = {200,100}, % \ll
8505    "1D = {100,200}, % \gg
8506    "1E = {300,100}, % \prec
8507    "1F = {100,300}, % \succ
8508    "20 = {100,200}, % \leftarrow
8509    "21 = {200,100}, % \rightarrow
8510    "22 = {100,100}, % \uparrow
8511    "23 = {100,100}, % \downarrow
8512    "24 = {100,100}, % \leftrightarrow
8513    "25 = {100,100}, % \nearrow
8514    "26 = {100,100}, % \searrow
8515    "27 = {100,100}, % \simeq
8516    "28 = {100,100}, % \Leftarrow
8517    "29 = {100,100}, % \Rightarrow
8518    "2A = {100,100}, % \Uparrow
8519    "2B = {100,100}, % \Downarrow
8520    "2C = {100,100}, % \Leftrightarrow
8521    "2D = {100,100}, % \nwarrow
8522    "2E = {100,100}, % \swarrow
8523    "2F = {    ,100}, % \propto
8524    "30 = {    ,400}, % \prime
8525    "31 = {100,100}, % \infty
8526    "32 = {150,100}, % \in
8527    "33 = {100,150}, % \ni
8528    "34 = {100,100}, % \triangle, \bigtriangleup
8529    "35 = {100,100}, % \bigtriangledown
8530    "38 = {    ,100}, % \forall
8531    "39 = {100,    }, % \exists
8532    "3A = {200,    }, % \neg
8533    "3E = {200,200}, % \top
8534    "3F = {200,200}, % \bot, \perp
8535    "5E = {100,200}, % \wedge
8536    "5F = {100,200}, % \vee
8537    "60 = {    ,300}, % \vdash
8538    "61 = {300,    }, % \dashv
8539    "62 = {100,100}, % \lfloor
8540    "63 = {100,100}, % \rfloor
```

```
8541    "64 = {100,100}, % \lceil
8542    "65 = {100,100}, % \rceil
8543    "66 = {150,    }, % \lbrace
8544    "67 = {    ,150}, % \rbrace
8545    "68 = {400,    }, % \langle
8546    "69 = {    ,400}, % \rangle
8547    "6C = {100,100}, % \updownarrow
8548    "6D = {100,100}, % \Updownarrow
8549    "6E = {100,300}, % \, \backslash, \setminus
8550    "72 = {100,100}, % \nabla
8551    "79 = {200,200}, % \dagger
8552    "7A = {100,100}, % \ddagger
8553    "7B = {100,    }, % \mathparagraph
8554    "7C = {100,100}, % \clubsuit
8555    "7D = {100,100}, % \diamondsuit
8556    "7E = {100,100}, % \heartsuit
8557    "7F = {100,100}  % \spadesuit
```

Remaining slots in the source file.

```
8558    }
8559
```

We don't bother about 'largesymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
```

```
8560  ⟨/cmr⟩
8561  ⟨/cfg-t⟩
```

2.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
8562  ⟨*cfg-u⟩
```

Symbol font 'a'.

```
8563  ⟨*msa⟩
8564  \SetProtrusion
8565    [ name      = AMS-a ]
8566    { encoding = U,
8567      family   = msa  }
8568    {
8569    "05  =  {150,250},  % \centerdot
8570    "06  =  {100,100},  % \lozenge
8571    "07  =  { 50, 50},  % \blacklozenge
8572    "08  =  { 50, 50},  % \circlearrowright
8573    "09  =  { 50, 50},  % \circlearrowleft
8574    "0A  =  {100,100},  % \rightleftharpoons
8575    "0B  =  {100,100},  % \leftrightharpoons
8576    "0D  =  {-50,200},  % \Vdash
8577    "0E  =  {-50,200},  % \Vvdash
8578    "0F  =  {-70,150},  % \vDash
8579    "10  =  {100,150},  % \twoheadrightarrow
8580    "11  =  {100,150},  % \twoheadleftarrow
8581    "12  =  { 50,100},  % \leftleftarrows
8582    "13  =  { 50, 80},  % \rightrightarrows
8583    "14  =  {120,120},  % \upuparrows
8584    "15  =  {120,120},  % \downdownarrows
8585    "16  =  {200,200},  % \upharpoonright
8586    "17  =  {200,200},  % \downharpoonright
8587    "18  =  {200,200},  % \upharpoonleft
8588    "19  =  {200,200},  % \downharpoonleft
8589    "1A  =  { 80,100},  % \rightarrowtail
8590    "1B  =  { 80,100},  % \leftarrowtail
```

```
8591    "1C  =  {  50,  50},   % \leftrightarrows
8592    "1D  =  {  50,  50},   % \rightleftarrows
8593    "1E  =  {250,    },   % \Lsh
8594    "1F  =  {    ,250},   % \Rsh
8595    "20  =  {100,100},   % \rightsquigarrow
8596    "21  =  {100,100},   % \leftrightsquigarrow
8597    "22  =  {100, 50},   % \looparrowleft
8598    "23  =  { 50,100},   % \looparrowright
8599    "24  =  { 50, 80},   % \circeq
8600    "25  =  {    ,100},   % \succsim
8601    "26  =  {    ,100},   % \gtrsim
8602    "27  =  {    ,100},   % \gtrapprox
8603    "28  =  {150, 50},   % \multimap
8604    "2B  =  {100,150},   % \doteqdot
8605    "2C  =  {100,150},   % \triangleq
8606    "2D  =  {100, 50},   % \precsim
8607    "2E  =  {100, 50},   % \lesssim
8608    "2F  =  { 50, 50},   % \lessapprox
8609    "30  =  {100, 50},   % \eqslantless
8610    "31  =  { 50, 50},   % \eqslantgtr
8611    "32  =  {100, 50},   % \curlyeqprec
8612    "33  =  { 50,100},   % \curlyeqsucc
8613    "34  =  {100, 50},   % \preccurlyeq
8614    "36  =  { 50,    },   % \leqslant
8615    "38  =  {    , 50},   % \backprime
8616    "39  =  {250,250},   % \dabar@ : the dash bar in \dash(left,right)arrow
8617    "3C  =  { 50,100},   % \succcurlyeq
8618    "3E  =  {    , 50},   % \geqslant
8619    "40  =  {    , 50},   % \sqsubset
8620    "41  =  { 50,    },   % \sqsupset
8621    "42  =  {    ,150},   % \vartriangleright, \rhd
8622    "43  =  {150,    },   % \vartriangleleft, \lhd
8623    "44  =  {    ,100},   % \trianglerighteq, \unrhd
8624    "45  =  {100,    },   % \trianglelefteq, \unlhd
8625    "46  =  {100,100},   % \bigstar
8626    "48  =  { 50, 50},   % \blacktriangledown
8627    "49  =  {    ,100},   % \blacktriangleright
8628    "4A  =  {100,    },   % \blacktriangleleft
8629    "4B  =  {    ,150},   % \dashrightarrow (the arrow)
8630    "4C  =  {150,    },   % \dashleftarrow
8631    "4D  =  { 50, 50},   % \vartriangle
8632    "4E  =  { 50, 50},   % \blacktriangle
8633    "4F  =  { 50, 50},   % \triangledown
8634    "50  =  { 50, 50},   % \eqcirc
8635    "56  =  {    ,150},   % \Rrightarrow
8636    "57  =  {150,    },   % \Lleftarrow
8637    "58  =  {100,300},   % \checkmark
8638    "5C  =  { 50, 50},   % \angle
8639    "5D  =  { 50, 50},   % \measuredangle
8640    "5E  =  { 50, 50},   % \sphericalangle
8641    "5F  =  {    , 50},   % \varpropto
8642    "60  =  {100,100},   % \smallsmile
8643    "61  =  {100,100},   % \smallfrown
8644    "62  =  { 50,    },   % \Subset
8645    "63  =  {    , 50},   % \Supset
8646    "66  =  {150,150},   % \curlywedge
8647    "67  =  {150,150},   % \curlyvee
8648    "68  =  { 50,150},   % \leftthreetimes
8649    "69  =  {100, 50},   % \rightthreetimes
8650    "6C  =  { 50, 50},   % \bumpeq
8651    "6D  =  { 50, 50},   % \Bumpeq
8652    "6E  =  {100,    },   % \lll
8653    "6F  =  {    ,100},   % \ggg
8654    "70  =  { 50,100},   % \ulcorner
8655    "71  =  {100, 50},   % \urcorner
```

```
8656        "75  =  {150,200},   % \dotplus
8657        "76  =  { 50,100},   % \backsim
8658        "78  =  { 50,100},   % \llcorner
8659        "79  =  {100, 50},   % \lrcorner
8660        "7C  =  {100,100},   % \intercal
8661        "7D  =  { 50, 50},   % \circledcirc
8662        "7E  =  { 50, 50},   % \circledast
8663        "7F  =  { 50, 50}    % \circleddash
```

Remaining slots in the source file.

```
8664      }
8665
8666  ⟨/msa⟩
```

Symbol font 'b'.

```
8667  ⟨*msb⟩
8668  \SetProtrusion
8669    [ name      = AMS-b ]
8670    { encoding = U,
8671      family   = msb   }
8672    {
8673        A  =  { 50, 50},   % \mathbb
8674        C  =  { 50, 50},
8675        G  =  {    , 50},
8676        L  =  {    , 50},
8677        P  =  {    , 50},
8678        R  =  {    , 50},
8679        T  =  {    , 50},
8680        V  =  { 50, 50},
8681        X  =  { 50, 50},
8682        Y  =  { 50, 50},
8683      "00  =  { 50, 50},   % \lvertneqq
8684      "01  =  { 50, 50},   % \gvertneqq
8685      "02  =  { 50, 50},   % \nleq
8686      "03  =  { 50, 50},   % \ngeq
8687      "04  =  {100, 50},   % \nless
8688      "05  =  { 50,150},   % \ngtr
8689      "06  =  {100, 50},   % \nprec
8690      "07  =  { 50,150},   % \nsucc
8691      "08  =  { 50, 50},   % \lneqq
8692      "09  =  { 50, 50},   % \gneqq
8693      "0A  =  {100,100},   % \nleqslant
8694      "0B  =  {100,100},   % \ngeqslant
8695      "0C  =  {100, 50},   % \lneq
8696      "0D  =  { 50,100},   % \gneq
8697      "0E  =  {100, 50},   % \npreceq
8698      "0F  =  { 50,100},   % \nsucceq
8699      "10  =  { 50,    },   % \precnsim
8700      "11  =  { 50, 50},   % \succnsim
8701      "12  =  { 50, 50},   % \lnsim
8702      "13  =  { 50, 50},   % \gnsim
8703      "14  =  { 50, 50},   % \nleqq
8704      "15  =  { 50, 50},   % \ngeqq
8705      "16  =  { 50, 50},   % \precneqq
8706      "17  =  { 50, 50},   % \succneqq
8707      "18  =  { 50, 50},   % \precnapprox
8708      "19  =  { 50, 50},   % \succnapprox
8709      "1A  =  { 50, 50},   % \lnapprox
8710      "1B  =  { 50, 50},   % \gnapprox
8711      "1C  =  {150,200},   % \nsim
8712      "1D  =  { 50, 50},   % \ncong
8713      "1E  =  {100,150},   % \diagup
8714      "1F  =  {100,150},   % \diagdown
8715      "20  =  {100, 50},   % \varsubsetneq
8716      "21  =  { 50,100},   % \varsupsetneq
```

```
8717      "22 = {100, 50},   % \nsubseteqq
8718      "23 = { 50,100},   % \nsupseteqq
8719      "24 = {100, 50},   % \subsetneqq
8720      "25 = { 50,100},   % \supsetneqq
8721      "26 = {100, 50},   % \varsubsetneqq
8722      "27 = { 50,100},   % \varsupsetneqq
8723      "28 = {100, 50},   % \subsetneq
8724      "29 = { 50,100},   % \supsetneq
8725      "2A = {100, 50},   % \nsubseteq
8726      "2B = { 50,100},   % \nsupseteq
8727      "2C = { 50,100},   % \nparallel
8728      "2D = {100,150},   % \nmid
8729      "2E = {150,150},   % \nshortmid
8730      "2F = {100,100},   % \nshortparallel
8731      "30 = {   ,150},   % \nvdash
8732      "31 = {   ,150},   % \nVdash
8733      "32 = {   ,100},   % \nvDash
8734      "33 = {   ,100},   % \nVDash
8735      "34 = {   ,100},   % \ntrianglerighteq
8736      "35 = {100,   },   % \ntrianglelefteq
8737      "36 = {100,   },   % \ntriangleleft
8738      "37 = {   ,100},   % \ntriangleright
8739      "38 = {100,200},   % \nleftarrow
8740      "39 = {100,200},   % \nrightarrow
8741      "3A = {100,100},   % \nLeftarrow
8742      "3B = { 50,100},   % \nRightarrow
8743      "3C = {100,100},   % \nLeftrightarrow
8744      "3D = {100,200},   % \nleftrightarrow
8745      "3E = { 50, 50},   % \divideontimes
8746      "3F = { 50, 50},   % \varnothing
8747      "60 = {200,   },   % \Finv
8748      "61 = {   , 50},   % \Game
8749      "68 = {100,100},   % \eqsim
8750      "69 = { 50,   },   % \beth
8751      "6A = { 50,   },   % \gimel
8752      "6B = {150,   },   % \daleth
8753      "6C = {200,   },   % \lessdot
8754      "6D = {   ,200},   % \gtrdot
8755      "6E = {100,200},   % \ltimes
8756      "6F = {150,100},   % \rtimes
8757      "70 = { 50,100},   % \shortmid
8758      "71 = { 50, 50},   % \shortparallel
8759      "72 = {200,300},   % \smallsetminus
8760      "73 = {100,200},   % \thicksim
8761      "74 = { 50,100},   % \thickapprox
8762      "75 = { 50, 50},   % \approxeq
8763      "76 = { 50,100},   % \succapprox
8764      "77 = { 50, 50},   % \precapprox
8765      "78 = {100,100},   % \curvearrowleft
8766      "79 = { 50,150},   % \curvearrowright
8767      "7A = { 50,200},   % \digamma
8768      "7B = {100, 50},   % \varkappa
8769      "7F = {200,   }    % \backepsilon
```

Remaining slots in the source file.

```
8770    }
8771
8772  ⟨/msb⟩
```

2.8.8 Euler

Euler Roman font (package euler).

```
8773  ⟨*eur⟩
8774  \SetProtrusion
```

```
8775    [ name     = euler ]
8776    { encoding = U,
8777      family   = eur  }
8778    {
8779      "01  =  {100,100},
8780      "03  =  {100,150},
8781      "06  =  {   ,100},
8782      "07  =  {100,150},
8783      "08  =  {100,100},
8784      "0A  =  {100,100},
8785      "0B  =  {   , 50},
8786      "0C  =  {   ,100},
8787      "0D  =  {100,100},
8788      "0E  =  {   ,100},
8789      "0F  =  {100,100},
8790      "10  =  {100,100},
8791      "13  =  {   ,100},
8792      "14  =  {   ,100},
8793      "15  =  {   , 50},
8794      "16  =  {   , 50},
8795      "17  =  { 50,100},
8796      "18  =  { 50,100},
8797      "1A  =  {   , 50},
8798      "1B  =  {   , 50},
8799      "1C  =  { 50,100},
8800      "1D  =  { 50,100},
8801      "1E  =  { 50,100},
8802      "1F  =  { 50,100},
8803      "20  =  {   , 50},
8804      "21  =  {   , 50},
8805      "22  =  { 50,100},
8806      "24  =  {   , 50},
8807      "27  =  { 50,100},
8808        1  =  {100,100},
8809        7  =  { 50,100},
8810      "3A  =  {300,500},
8811      "3B  =  {200,400},
8812      "3C  =  {200,100},
8813      "3D  =  {200,200},
8814      "3E  =  {100,200},
8815        A  =  {   ,100},
8816        D  =  {   , 50},
8817        J  =  { 50,   },
8818        K  =  {   , 50},
8819        L  =  {   , 50},
8820        Q  =  {   , 50},
8821        T  =  { 50,   },
8822        X  =  { 50, 50},
8823        Y  =  { 50,   },
8824        h  =  {   , 50},
8825        k  =  {   , 50}
8826    }
8827
```

Extended by the eulervm package.

```
8828  \SetProtrusion
8829    [ name     = euler-vm,
8830      load     = euler ]
8831    { encoding = U,
8832      family   = zeur  }
8833    {
8834      "28  =  {100,200},
8835      "29  =  {100,200},
8836      "2A  =  {100,150},
8837      "2B  =  {100,150},
```

```
8838        "2C  =  {200,300},
8839        "2D  =  {200,300},
8840        "2E  =  {   ,100},
8841        "2F  =  {100,   },
8842        "3F  =  {150,150},
8843        "5B  =  {   ,100},
8844        "5E  =  {100,100},
8845        "5F  =  {100,100},
8846        "80  =  {   , 50},
8847        "81  =  {200,250},
8848        "82  =  {100,200}
8849      }
8850
```
8851 ⟨/eur⟩

Euler Script font (eucal).

8852 ⟨*eus⟩
```
8853 \SetProtrusion
8854   [ name     = euscript ]
8855   { encoding = U,
8856     family   = eus  }
8857   {
8858      A  =  {100,100},
8859      B  =  { 50,100},
8860      C  =  { 50, 50},
8861      D  =  { 50,100},
8862      E  =  { 50,100},
8863      F  =  { 50,   },
8864      G  =  { 50,   },
8865      H  =  {   ,100},
8866      K  =  {   , 50},
8867      L  =  {   ,150},
8868      M  =  {   , 50},
8869      N  =  {   , 50},
8870      O  =  { 50, 50},
8871      P  =  { 50, 50},
8872      T  =  {   ,100},
8873      U  =  {   , 50},
8874      V  =  { 50, 50},
8875      W  =  { 50, 50},
8876      X  =  { 50, 50},
8877      Y  =  { 50,   },
8878      Z  =  { 50,100},
8879     "00  =  {250,250},
8880     "18  =  {200,200},
8881     "3A  =  {200,150},
8882     "40  =  {   ,100},
8883     "5E  =  {100,100},
8884     "5F  =  {100,100},
8885     "66  =  { 50,   },
8886     "67  =  {   , 50},
8887     "6E  =  {200,200}
8888   }
8889
8890 \SetProtrusion
8891   [ name     = euscript-vm,
8892     load     = euscript ]
8893   { encoding = U,
8894     family   = zeus  }
8895   {
8896     "01  =  {600,600},
8897     "02  =  {200,200},
8898     "03  =  {200,200},
8899     "04  =  {200,200},
8900     "05  =  {150,150},
```

```
8901     "06   =   {200,200},
8902     "07   =   {200,200},
8903     "08   =   {100,100},
8904     "09   =   {100,100},
8905     "0A   =   {100,100},
8906     "0B   =   {100,100},
8907     "0C   =   {100,100},
8908     "0D   =   {100,100},
8909     "0E   =   {150,150},
8910     "0F   =   {100,100},
8911     "10   =   {150,150},
8912     "11   =   {100,100},
8913     "12   =   {150,100},
8914     "13   =   {100,150},
8915     "14   =   {150,100},
8916     "15   =   {100,150},
8917     "16   =   {200,100},
8918     "17   =   {100,200},
8919     "19   =   {150,150},
8920     "1A   =   {150,100},
8921     "1B   =   {100,150},
8922     "1C   =   {100,100},
8923     "1D   =   {100,100},
8924     "1E   =   {250,100},
8925     "1F   =   {100,250},
8926     "20   =   {150,200},
8927     "21   =   {150,200},
8928     "22   =   {150,150},
8929     "23   =   {150,150},
8930     "24   =   {100,200},
8931     "25   =   {150,150},
8932     "26   =   {150,150},
8933     "27   =   {100,100},
8934     "28   =   {100,100},
8935     "29   =   {100,150},
8936     "2A   =   {100,100},
8937     "2B   =   {100,100},
8938     "2C   =   {100,100},
8939     "2D   =   {150,150},
8940     "2E   =   {150,150},
8941     "2F   =   {100,100},
8942     "30   =   {100,100},
8943     "31   =   {100,100},
8944     "32   =   {100,100},
8945     "33   =   {100,100},
8946     "34   =   {100,100},
8947     "35   =   {100,100},
8948     "3E   =   {150,150},
8949     "3F   =   {150,150},
8950     "60   =   {   ,200},
8951     "61   =   {200,   },
8952     "62   =   {100,100},
8953     "63   =   {100,100},
8954     "64   =   {100,100},
8955     "65   =   {100,100},
8956     "68   =   {300,   },
8957     "69   =   {   ,300},
8958     "6C   =   {100,100},
8959     "6D   =   {100,100},
8960     "6F   =   {100,100},
8961     "72   =   {100,100},
8962     "73   =   {200,100},
8963     "76   =   {   ,100},
8964     "77   =   {100,   },
8965     "78   =   { 50, 50},
```

```
8966      "79  =  {100,100},
8967      "7A  =  {100,100},
8968      "7D  =  {150,150},
8969      "7E  =  {100,100},
8970      "A8  =  {100,100},
8971      "A9  =  {100,100},
8972      "AB  =  {200,200},
8973      "BA  =  {    ,200},
8974      "BB  =  {    ,200},
8975      "BD  =  {200,200},
8976      "DE  =  {200,200}
8977    }
8978
8979 ⟨/eus⟩
```

Euler Fraktur font (eufrak).

```
8980 ⟨*euf⟩
8981 \SetProtrusion
8982    [ name     = mathfrak ]
8983    { encoding = U,
8984      family   = euf  }
8985    {
8986       A  =  {   , 50},
8987       B  =  {   , 50},
8988       C  =  { 50, 50},
8989       D  =  {   , 80},
8990       E  =  { 50,   },
8991       G  =  {   , 50},
8992       L  =  {   , 80},
8993       O  =  {   , 50},
8994       T  =  {   , 80},
8995       X  =  { 80, 50},
8996       Z  =  { 80, 50},
8997       b  =  {   , 50},
8998       c  =  {   , 50},
8999       k  =  {   , 50},
9000       p  =  {   , 50},
9001       q  =  { 50,   },
9002       v  =  {   , 50},
9003       w  =  {   , 50},
9004       x  =  {   , 50},
9005       1  =  {100,100},
9006       2  =  { 80, 80},
9007       3  =  { 80, 50},
9008       4  =  { 80, 50},
9009       7  =  { 50, 50},
9010      "12  =  {500,500},
9011      "13  =  {500,500},
9012       !  =  {   ,200},
9013       '  =  {200,300},
9014       (  =  {200,   },
9015       )  =  {   ,200},
9016       *  =  {200,200},
9017       +  =  {200,250},
9018       -  =  {200,200},
9019     {,} =  {300,300},
9020       .  =  {400,400},
9021     {=} =  {200,200},
9022       :  =  {   ,200},
9023       ;  =  {   ,200},
9024       ]  =  {   ,200}
9025    }
9026
9027 ⟨/euf⟩
9028 ⟨/cfg-u⟩
```

2.8.9   Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages `eurosans`, `europs`), ITC Euro fonts (package `euroitc`) and `marvosym`[13]). The `euroitc` settings are hidden in the package itself (1.3.7) for 'free software' compliance reasons. (Not quite sure whether this is what Karl really had in mind . . .)

```
9029  ⟨*cfg-e⟩
9030  \SetProtrusion
9031  ⟨zpeu⟩   { encoding = U,
9032  ⟨mvs⟩    { encoding = {OT1,U},
9033  ⟨zpeu⟩      family   = zpeu }
9034  ⟨mvs⟩      family   = mvs }
9035    {
9036  ⟨zpeu⟩      E = {50, }
9037  ⟨mvs⟩      164 = {50,50},   % \EUR
9038  ⟨mvs⟩      068 = {50,-100}  % \EURdig
9039    }
9040
9041  ⟨*zpeu⟩
9042  \SetProtrusion
9043    { encoding = U,
9044      family   = zpeu,
9045      shape    = it* }
9046    {
9047      E = {100,-50}
9048    }
9049
9050  \SetProtrusion
9051    { encoding = U,
9052      family   = {zpeus,eurosans} }
9053    {
9054      E = {100,50}
9055    }
9056
9057  \SetProtrusion
9058    { encoding = U,
9059      family   = {zpeus,eurosans},
9060      shape    = it* }
9061    {
9062      E = {200, }
9063    }
9064
9065  ⟨/zpeu⟩
9066  ⟨/cfg-e⟩
```

2.9   Interword spacing

Default unit is space.

```
9067  ⟨*m-t|cmr⟩
9068  %%% -----------------------------------------------------------------------
9069  %%% INTERWORD SPACING
9070
9071  ⟨/m-t|cmr⟩
9072  ⟨*m-t⟩
9073  \SetExtraSpacing
9074    [ name = default ]
9075    { encoding = {OT1,T1,LY1,OT4,QX,T5} }
9076    {
```

These settings are only a first approximation. The following reasoning is from a

---

13   Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

2	6	7	5	3	4	1

Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei

mail from *Ulrich Dirr*, who also provided the sample in figure 1. I do not claim to
have coped with the task.

'The idea is – analog to the tables for expansion and protrusion – to have
tables for optical reduction/expansion of spaces in dependence of the actual
character so that the distance between words is optically equal.

When reducing distances the (weighting) order is:

- after commas

9077   `{,} = { ,-500,500},`

- in front of capitals which have optical more room on their left side, e.g., 'A',
  'J', 'T', 'V', 'W', and 'Y' [this is not yet possible – RS]

- in front of capitals which have circle/oval shapes on their left side, e.g., 'C',
  'G', 'O', and 'Q' [ditto – RS]

- after 'r' (because of the bigger optical room on the righthand side)

9078   `r = { ,-300,300},`

- [before or] after lowercase characters with ascenders

9079   `b = { ,-200,200},`
9080   `d = { ,-200,200},`
9081   `f = { ,-200,200},`
9082   `h = { ,-200,200},`
9083   `k = { ,-200,200},`
9084   `l = { ,-200,200},`
9085   `t = { ,-200,200},`

- [before or] after lowercase characters with x-height plus descender with
  additional optical space, e.g., 'v', or 'w'

9086   `c = { ,-100,100},`
9087   `p = { ,-100,100},`
9088   `v = { ,-100,100},`
9089   `w = { ,-100,100},`
9090   `z = { ,-100,100},`
9091   `x = { ,-100,100},`
9092   `y = { ,-100,100},`

- [before or] after lowercase characters with x-height plus descender without
  additional optical space

9093   `i = { , 50, -50},`
9094   `m = { , 50, -50},`
9095   `n = { , 50, -50},`
9096   `u = { , 50, -50},`

- after colon and semicolon

9097   `: = { ,200,-200},`
9098   `; = { ,200,-200},`

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

```
9099          .  = { ,250,-250},
9100          !  = { ,250,-250},
9101          ?  = { ,250,-250}
```

The order has to be reversed when enlarging is needed.'

```
9102    }
9103
9104 ⟨/m-t⟩
```

Questions are:

- Is the result really better?

- Is it overdone? (Try with a factor < 1000.)

- Should the first parameter also be used? (Probably.)

- What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTEX bug with spacing in combination with a non-zero \spaceskip (reported by *Axel Berger*):

```
\parfillskip0pt
\rightskip0pt plus 1em
\spaceskip\fontdimen2\font
 test test\par
\pdfadjustinterwordglue2
\stbscode\font`t=-50
 test test
\bye
```

Some more characters in T2A. [14]

```
9105 ⟨*cmr⟩
9106 \SetExtraSpacing
9107    [ name     = T2A,
9108      load     = default ]
9109    { encoding = T2A,
9110      family   = cmr }
9111    {
9112      \cyrg  = { ,-300,300},
9113      \cyrb  = { ,-200,200},
9114      \cyrk  = { ,-200,200},
9115      \cyrs  = { ,-100,100},
9116      \cyrr  = { ,-100,100},
9117      \cyrh  = { ,-100,100},
9118      \cyru  = { ,-100,100},
9119      \cyrt  = { , 50, -50},
9120      \cyrp  = { , 50, -50},
9121      \cyri  = { , 50, -50},
9122      \cyrishrt  = { , 50, -50},
9123    }
9124
```

2.9.1 Nonfrenchspacing

The following settings simulate \nonfrenchspacing (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the TEXbook:

---

14   Contributed by *Karl Karlsson*.

> 'If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [. . .] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.'

The 'extra space' (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i.e., 333.

```
9125 \SetExtraSpacing
9126   [ name    = nonfrench-cmr,
9127     load    = default,
9128     context = nonfrench ]
9129   { encoding = {OT1,T1,LY1,OT4,QX,T5},
9130     family  = cmr }
9131   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
  \sfcode`\. 3000
  \sfcode`\? 3000
  \sfcode`\! 3000
```

```
9132     . = {333,2000,-667},
9133     ? = {333,2000,-667},
9134     ! = {333,2000,-667},
```

```
  \sfcode`\: 2000
```

```
9135     : = {333,1000,-500},
```

```
  \sfcode`\; 1500
```

```
9136     ; = {   , 500,-333},
```

```
  \sfcode`\, 1250
```

```
9137   {,}= {   , 250,-200}
```

```
}
```

```
9138   }
9139
9140 ⟨/cmr⟩
```

`fontinst`, however, which is also used to create the `psnfss` font metrics, sets `\fontdimen 7` to 240 by default. Therefore, the fallback settings use this value for the first component.

```
9141 ⟨*m-t⟩
9142 \SetExtraSpacing
9143   [ name    = nonfrench-default,
9144     load    = default,
9145     context = nonfrench ]
9146   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
9147   {
9148     . = {240,2000,-667},
9149     ? = {240,2000,-667},
9150     ! = {240,2000,-667},
9151     : = {240,1000,-500},
9152     ; = {   , 500,-333},
9153   {,}= {   , 250,-200}
9154   }
9155
```

Empty settings to prevent spurious warnings.

```
9156 \SetExtraSpacing
9157    [ name = empty ]
9158    { encoding = {TS1} }
9159    { }
9160
```

2.10 Additional kerning

Default unit is 1 em.

```
9161 %%% ------------------------------------------------------------------------
9162 %%% ADDITIONAL KERNING
9163
```

A dummy list to be loaded when no context is active.

```
9164 \SetExtraKerning
9165    [ name = empty ]
9166    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
9167    { }
9168
```

2.10.1 French

The ratio of \fontdimen 2 to \fontdimen 6 varies for different fonts, so that either the kerning of the colon (which should be a space, i.e., \fontdimen 2) or that of the other punctuation characters (TEX's \thinspace, i.e., one sixth of \fontdimen 6) may be inaccurate, depending on which unit we choose (space or 1 em). For Times, for example, a thin space would be 665. I don't know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a space. (Wikipedia [15] claims it should be a quarter of an em, which seems too much to me; then again, it also says that this *was* a thin space in French typography.)

```
9169 \SetExtraKerning
9170    [ name      = french-default,
9171      context   = french,
9172      unit      = space   ]
9173    { encoding = {OT1,T1,LY1} }
9174    {
9175      :  = {1000,}, % = \fontdimen2
9176      ;  = {500, }, % ~ \thinspace
9177      !  = {500, },
9178      ?  = {500, }
9179    }
9180
```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTEX.

```
9181 \SetExtraKerning
9182    [ name      = french-guillemets,
9183      context   = french-guillemets,
9184      load      = french-default,
9185      unit      = space   ]
9186    { encoding = {T1,LY1} }
9187    {
9188     \guillemotleft  = { ,800}, % = 0.8\fontdimen2
9189     \guillemotright = {800, }
9190    }
9191
```

---

15 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

```
9192 \SetExtraKerning
9193   [ name      = french-guillemets-OT1,
9194     context  = french-guillemets,
9195     load     = french-default,
9196     unit     = space   ]
9197   { encoding = OT1     }
9198   { }
9199
```

2.10.2 Turkish

```
9200 \SetExtraKerning
9201   [ name      = turkish,
9202     context  = turkish ]
9203   { encoding = {OT1,T1,LY1} }
9204   {
9205     :  = {167, }, % = \thinspace
9206     !  = {167, },
9207    {=} = {167, }
9208   }
9209
9210 ⟨/m-t⟩
9211 ⟨/config⟩
```

3   OpenType configuration files

These are the configuration files for the following OpenType fonts: [16]

- Latin Modern Roman
- New Computer Modern [17]
- Charis SIL
- EB Garamond
- Palatino [18]

The settings are typeset in the respective font.

3.1   Character inheritance

OpenType fonts may differ considerably in how complete their arsenal of glyphs is. Therefore, each font family should have their own inheritance settings.

```
9212
9213 %%% ----------------------------------------------------------------------
9214 %%% INHERITANCE
9215
9216 %% for xetex (EU1) and luatex (EU2), resp. both (TU)
```

3.1.1   Latin Modern Roman/New Computer Modern

```
9217 ⟨*LatinModernRoman|NewComputerModern⟩
9218 \DeclareCharacterInheritance
9219   { encoding = {TU,EU1,EU2},
9220 ⟨LatinModernRoman⟩    family  = Latin Modern Roman }
9221 ⟨NewComputerModern⟩    family  = {New Computer Modern} }
9222   {
9223     A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǽ,Ä,Ą,Å,Ấ,Ầ,Ẫ,Ẩ,Ậ,Ắ,Ằ,Ẵ,Ặ,
9224 ⟨LatinModernRoman⟩        A % Greek
9225 ⟨NewComputerModern⟩       A,A,Ą,Ą,Ă,Ā  % Ά,Ᾰ,Ᾱ,Ὰ,Ά,Ᾱ,Ᾱ,ΑΙ,Ἀ,Ἄ,Ἂ,Ἆ,Ἁ,Ἅ,Ἃ,Ἇ,Ἀ,  % Greek
9226        },
9227     Æ = {Ǽ},
9228     B = {ℬ,
9229        B},  % Greek
9230     C = {Ç,Ć,Ĉ,Ċ,Č},
9231     D = {Đ,Ď,Ḍ,Ḏ,Ḏ},
9232     E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,È,Ẹ,Ẻ,Ẽ,Ế,Ề,Ễ,Ể,Ệ,
9233        E},  % Greek
9234 ⟨NewComputerModern⟩ (l)E = {E,Ἐ,Ἔ,Ἒ,Ἒ,Ἓ,Ἕ,Ὲ,Έ}, % Greek accents fully protruded left
9235     G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ},
9236     H = {Ĥ,Ħ,Ḥ,Ḧ,Ḩ,
9237 ⟨LatinModernRoman⟩        H  % Greek
9238 ⟨NewComputerModern⟩       H,Ḥ  % Greek
9239        },
9240 ⟨NewComputerModern⟩ (l)H = {Ἠ,Ἤ,Ἢ,Ἦ,Ἧ,Ἦ,Ἡ,Ἥ,Ἣ,Ἣ,Ἧ,Ὴ,Ή,Η͂,Η͂Ι,ΗΙ,Ἤ}, % Greek accents fully protruded left
9241 ⟨NewComputerModern⟩ %(l)/uni1FCC.alt = {/uni1F98.alt},
9242     I = {Ì,Í,Î,Ï,Ĩ,Ī,Ĭ,Į,Ị,Ỉ,Ị,Ị,
9243 ⟨LatinModernRoman⟩        I  % Greek
9244 ⟨NewComputerModern⟩       I,Ĭ,Ī  % Greek
```

---

16   This is file microtype-utf.dtx.
17   These settings have been contributed by *Antonis Tsolomitis*.
18   These settings have been contributed by *Loren B. Davis*.

```
9245        },
9246 ⟨NewComputerModern⟩ (l)I = {ʹI,ʼI,ˮI,ˊI,ˉI,ˈI,ˮI,ˋI,ˈI}, % Greek
9247        J = {Ĵ},
9248        K = {Ķ,
9249           K}, % Greek
9250        L = {Ĺ,Ļ,Ł,Ḽ}, % Ł,L,Ḽ
9251        M = {M}, % Greek
9252        N = {Ñ,Ń,Ņ,Ň,Ǹ,Ṉ,
9253           N}, % Greek
9254        O = {Ò,Ó,Ô,Õ,Ö,Ø,Ō,Ŏ,Ő,Ǫ,Ơ,Ǒ,Ǿ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ỡ,Ở,Ợ,
9255           O}, % Greek
9256 ⟨NewComputerModern⟩ (l)O = {O,ʼO,ˊO,ˮO,ˋO,ˮO,ˮO,O}, % Greek accents except O that has in-
     dep. protrusion numbers (below)
9257        P = {P}, % Greek
9258 ⟨NewComputerModern⟩ (l)P = {ʻP}, % Greek accents fully protruded left
9259        R = {Ŕ,Ŗ,Ř,Ȑ,Ŗ,Ṙ,Ṟ},
9260        S = {Ś,Ŝ,Ş,Š,Ș,Ṣ},
9261        T = {Ţ,Ť,Ţ,Ț,Ṯ,
9262           T}, % Greek
9263        U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ư,Ǜ,Ụ,Ủ,Ứ,Ừ,Ǔ,Ữ,Ự},
9264        W = {Ŵ,Ẁ,Ẃ,Ẅ},
9265        X = {X}, % Greek
9266        Y = {Ý,Ŷ,Ÿ,Y,Ỷ,Ỹ},
9267 ⟨NewComputerModern⟩      Υ = {Ϋ,Ῠ,Ῡ},
9268 ⟨NewComputerModern⟩ (l)Υ = {Υ,ˋΥ,ˊΥ,ˮΥ,ˮΥ,ˮΥ,ˮΥ,Υ},
9269        Z = {Ź,Ż,Ž,Ẓ,
9270           Z}, % Greek
9271        a = {à,á,â,ã,ä,å,ā,ă,ą,ả,ằ,ạ,ả,ấ,ầ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ},
9272        æ = {ǽ},
9273        c = {ç,ć,ĉ,ċ,č},
9274        d = {đ,ď,ḏ},
9275        e = {è,é,ê,ë,ē,ĕ,ě,ę,ẹ,ẻ,ẽ,ế,ề,ể,ễ,ệ,ê},
9276        f = {ﬀ}, % Unicode 64256, glyph name in Latin Modern Roman: /f_f ; in New Com-
     puter Modern: /ﬀ
9277        g = {ĝ,ğ,ġ,ģ,ǧ,ǵ},
9278        h = {ĥ,ħ,ḥ,ḧ,ḩ},
9279        i = {ì,í,î,ï,ī,ĭ,į,ı,ỉ,ǐ,ị},
9280        j = {ĵ},
9281        k = {ķ},
9282        l = {ĺ,ļ,ł,ḷ,ḻ}, % ł,l ·
9283        n = {ñ,ń,ņ,ň,ṅ,ṇ},
9284        o = {ò,ó,ô,õ,ö,ø,ō,ŏ,ő,ơ,ǫ,ǿ,ọ,ỏ,ố,ồ,ổ,ỗ,ộ,ớ,ờ,ỡ,ở,ợ
9285 ⟨NewComputerModern⟩      ,o,ό,ὸ,ό,ῶ,ὸ,ὅ,ὸ,ό % Greek
9286        },
9287        r = {ŕ,ŗ,ř,ȑ,ŗ,ṙ,ṟ},
9288        s = {ś,ŝ,ş,š,ș,ṣ},
9289        t = {ţ,ť,ţ,ț,ẗ}, % ŧ
9290        u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ư,ǜ,ụ,ủ,ứ,ừ,ǔ,ữ,ự},
9291        w = {ŵ,ẁ,ẃ,ẅ},
9292        y = {ý,ŷ,ÿ,ỳ,ỵ,ỷ,ỹ},
9293        z = {ź,ż,ž,ẓ},
9294 ⟨*NewComputerModern⟩
9295        α = {ά,ὰ,ᾶ,ᾳ,ἀ,ἁ,ἄ,ἂ,ᾄ,ᾂ,ἆ,ᾆ,ᾷ,ᾴ,ᾲ,ᾁ},
9296        ε = {έ,ὲ,ἐ,ἑ,ἔ,ἒ,ἓ,ἕ},
9297        η = {ή,ὴ,ῆ,ῃ,ἠ,ἡ,ἤ,ἢ,ᾔ,ᾒ,ἦ,ᾖ,ῇ,ῄ,ῂ,ᾗ},
9298        ι = {ί,ὶ,ῖ,ἰ,ἱ,ἴ,ἲ,ἶ,ἷ,ί},
9299        ï = {ΐ,ῒ,ῗ,ϊ,ϊ,ἳ,ἵ,ῐ,ῑ,ΐ},
9300        υ = {ύ,ὺ,ῦ,ὐ,ὑ,ὔ,ὒ,ὖ,ὗ,ῠ,ῡ,ΰ},
9301        ω = {ώ,ὼ,ῶ,ῳ,ὠ,ὡ,ὤ,ὢ,ὦ,ᾦ,ᾧ,ῷ,ῴ,ῲ,ᾠ,ᾡ,ᾤ},
9302 ⟨/NewComputerModern⟩
9303      }

9304 ⟨/LatinModernRoman|NewComputerModern⟩
```

3.1.2 Charis SIL

```
9305 ⟨*CharisSIL⟩
9306 \DeclareCharacterInheritance
9307   { encoding = {TU,EU1,EU2},
9308     family   = Charis SIL }

9309   { A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ⱥ,ᴀ,
9310         А,Ӑ,Ӓ}, % Cyrillic
9311     Æ = {Ǽ,
9312         Ǣ,Ӕ}, % Cyrillic
9313     B = {Ḃ,Ḅ,Ḇ,
9314         В}, % Cyr
9315     C = {Ç,Ć,Ĉ,Ċ,Č,
9316         С,Ҫ}, % Cyr
9317     D = {Ď,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ,Đ},
9318     E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ȩ,Ḕ,Ḗ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,Ể,Ễ,Ệ,
9319         Е,Ѐ,Ё,Ӗ}, % Cyr
9320     F = {Ḟ},
9321     G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ,Ḡ,ɢ},
9322     H = {Ĥ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ,
9323         Н,Ң,Ҥ,Ҋ,н}, % Cyr
9324     I = {Ì,Í,Î,Ĩ,Ï,Į,Ĭ,Ī,İ,Ȉ,Ȋ,Ḭ,Ḯ,Ỉ,Ị,
9325         І,Ї,И,Й}, % Cyr
9326     J = {Ĵ,
9327         Ј}, % Cyr
9328     K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ,
9329         К,Ќ,Қ,Ҡ,Ҟ,Ҝ,Қ,ᴋ}, % Cyr
9330     L = {Ĺ,Ļ,Ľ,Ḷ,Ḹ,Ḻ,Ḽ}, % L·
9331     M = {Ḿ,Ṁ,Ṃ,
9332         М,Ӎ}, % Cyr
9333     N = {Ñ,Ń,Ņ,Ň,Ṅ,Ṇ,Ṉ,Ṋ,ɴ,
9334         И,Й,Ѝ,Ҋ,Ӥ,Й}, % Cyr
9335     O = {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Ǒ,Ǫ,Ǭ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ṍ,Ṏ,Ṑ,Ṓ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ,
9336         О,Ѳ,Ӧ,Ѳ,Ӫ, % Cyr
9337         Θ}, % Greek
9338     P = {Ṕ,Ṗ,
9339         Р,Ҏ},   % Cyr
9340     Q = {Ǫ}, % Cyr
9341     R = {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ},
9342     S = {Ś,Ŝ,Ş,Š,Ṡ,Ṣ,Ș,Ṥ,Ṧ,Ṩ,
9343         Ѕ}, % Cyr
9344     T = {Ţ,Ť,Ṭ,Ṫ,Ṱ,Ṯ,Ț,
9345         Т,Ҭ}, % Cyr
9346     U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
9347     V = {Ṽ,Ṿ},
9348     W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ,
9349         Ԝ}, % Cyr
9350     X = {Ẋ,Ẍ,
9351         Х,Ҳ,Ӿ,Ӽ},   % Cyr
9352     Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ,
9353         У,Ұ}, % Cyr
9354     Z = {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ},
9355     a = {à,á,â,ã,ä,å,ā,ă,ą,ǎ,ǟ,ǡ,ả,ấ,ầ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ,ặ, % a'
9356         а,ӑ,ӓ}, % Cyr
9357     æ = {ǽ,
9358         ӕ}, % Cyr
9359     b = {ḃ,ḅ,ḇ},
9360     c = {ç,ć,ĉ,ċ,č,ć,
9361         с,ҫ}, % Cyr
9362     d = {ď,ḋ,ḍ,ḏ,ḑ,ḓ},
9363     e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ȩ,ḕ,ḗ,ḙ,ḛ,ḝ,ẹ,ẻ,ẽ,ế,ề,ể,ễ,ệ,
9364         е,ѐ,ё,ӗ}, % Cyr
9365     f = {ḟ,ff}, % /f_f
```

```
9366        g  =  {ĝ,ğ,ġ,ġ,ǧ,ǵ,ḡ},
9367        h  =  {ĥ,ȟ,ḣ,ḥ,ḥ,ḧ,ḩ,ḫ,
9368              ɦ,ħ},  % Cyr
9369        i  =  {ı,ì,í,î,ĩ,ī,ĭ,į,ı,ḭ,ỉ,í,i,î,ĭ,
9370              i,ï},  % Cyr
9371        j  =  {ĵ,ǰ,
9372              ј},  % Cyr
9373        k  =  {ķ,ǩ,ḱ,ḳ,ḵ},
9374        l  =  {ĺ,ļ,ḽ,ḷ,ḹ,ḻ},  % ľ,l·
9375        m  =  {ḿ,ṁ,ṃ},
9376        n  =  {ñ,ń,ņ,ň,ǹ,ṅ,ṇ,ṉ,ṋ},  % 'n
9377        o  =  {ò,ó,ô,õ,ö,ō,ŏ,ő,ǒ,ǫ,ǭ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ố,ồ,ổ,ỗ,ộ,ớ,ờ,ở,ỡ,ợ,
9378              о,ө,ӧ,ѳ,ӫ},  % Cyr
9379        p  =  {ṕ,ṗ,
9380              р,ҏ},  % Cyr
9381        q  =  {q},  % Cyr
9382        r  =  {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
9383        s  =  {ś,ŝ,ş,š,ṡ,ṣ,ș,ṥ,ṧ,ṩ,
9384              ѕ},  % Cyr
9385        t  =  {ţ,ť,ṫ,ṭ,ṯ,ṱ,ẗ},  % ŧ
9386        u  =  {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự},
9387        v  =  {ṽ,ṿ},
9388        w  =  {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ,
9389              ԝ},   % Cyr
9390        x  =  {ẋ,ẍ,
9391              х},  % Cyr
9392        y  =  {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,ỵ,ỷ,ỹ,
9393              у,ў,ӯ,ӱ,ӳ},  % Cyr
9394        z  =  {ź,ż,ž,ẑ,ẓ,ẕ},
9395     % Cyrillic
9396        Ѓ  =  {Ѓ,Ғ,Ҕ,Ӻ,Ӷ},
9397        Ж  =  {Җ,Ӝ,Ӂ},
9398        З  =  {Ӟ,Ҙ},
9399        Л  =  {Ӆ},
9400        П  =  {Ҧ},
9401        У  =  {Ў,Ӯ,Ӱ,Ӳ},
9402        Ч  =  {Ҷ,Ҹ,Ӌ,Ӵ},
9403        Ы  =  {Ӹ},
9404        Ә  =  {Ӛ},
9405        Є  =  {Ҫ},
9406        г  =  {ѓ,ғ,ҕ,ӻ,ӷ},
9407        ж  =  {җ,ӝ,ӂ},
9408        з  =  {ҙ,ӟ},
9409        и  =  {й,ѝ,ӣ,ӣ,ӥ},
9410        к  =  {ќ,қ,ҝ,ҟ,ҡ,ӄ,ҡ},
9411        л  =  {ӆ},
9412        м  =  {ӎ},
9413        н  =  {ҥ,ӈ,ҧ,ӊ},
9414        п  =  {ҧ},
9415        т  =  {ҭ},
9416        х  =  {ӿ,ҳ},
9417        ч  =  {ҷ,ҹ,ӌ,ӵ},
9418        ш  =  {щ},
9419        ы  =  {ӹ},
9420        э  =  {ӭ},
9421        є  =  {ҫ},
9422        ә  =  {ӛ},
9423        ү  =  {ұ},
9424        Γ  =  {Γ},  % Greek
9425        Π  =  {Π},  % Greek
9426     % missing: tipa, math, symbols, ...
9427        }
```

9428 ⟨/CharisSIL⟩

3.1.3   EB Garamond

```
9429 ⟨*EBGaramond⟩
9430 \DeclareCharacterInheritance
9431   { encoding = {TU,EU1,EU2},
9432     family   = EBGaramond }
9433   {
9434     A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ǻ,Ȁ,Ȃ,Ȧ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ⱥ,Ǎ,/Adieresis.deu,
9435         A,Ӑ,Ӓ,                % Cyrillic
9436         Ά,Ἀ,Ἁ,Ἂ,Ἃ,Ἄ,Ἅ,Ἆ,Ἇ,Ᾰ,Ᾱ,Ὰ,Ά,ᾼ,ἈΙ,ἉΙ,ἊΙ,ἋΙ,ἌΙ,ἍΙ,ἎΙ,ἏΙ,Ᾰ,Ᾱ,Ὰ,Ά}, % Greek
9437 % (l)A  {Ά,Ἀ,Ἁ,Ἂ,Ἃ,Ἄ,Ἅ,Ἆ,Ἇ,Ᾰ,Ᾱ,Ὰ,Ά,ᾼ,Ᾰ,Ᾱ,Ὰ,Ά,Ᾰ,Ᾱ,Ὰ,Ά}, % (accents not protruded)
9438     B = {Ḃ,Ḅ,Ḇ,
9439         В,                   % Cyrillic
9440         Β},                  % Greek
9441     C = {Ç,Ć,Ĉ,Ċ,Č,Ḉ,
9442         С,Ҫ,Ҁ,               % Cyrillic
9443         C},                  % Roman numeral
9444     D = {Ď,Đ,Ð,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ,
9445         Ђ,Ѓ,                 % Cyrillic
9446         D},                  % Roman numeral
9447     E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ȇ,Ȩ,Ḕ,Ḗ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,
9448         Ѐ,Ё,Ӗ,Е,             % Cyrillic
9449         Ε},                  % Greek
9450 (l)E = {Έ,Ἐ,Ἑ,Ἒ,Ἓ,Ἔ,Ἕ,Ὲ,Έ},    % Greek (accents protruded)
9451     F = {Ḟ},
9452     G = {Ĝ,Ğ,Ġ,Ģ,Ǥ,Ǧ,Ǵ,Ḡ},
9453     H = {Ĥ,Ħ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ,Ḥ,Ħ,
9454         Н,Ң,Ҥ,Ҧ,  % Ҥ        % Cyrillic
9455         Η},                  % Greek
9456 (l)H = {Ή,Ἠ,Ἡ,Ἢ,Ἣ,Ἤ,Ἥ,Ἦ,Ἧ,Ὴ,Ή,ῌ,ἨΙ,ἩΙ,ἪΙ,ἫΙ,ἬΙ,ἭΙ,ἮΙ,ἯΙ},
9457     I = {Ì,Í,Î,Ï,Ĩ,Ī,Ĭ,Į,İ,Ǐ,Ȉ,Ȋ,Ḭ,Ḯ,Ỉ,Ị,
9458         Ӥ,Ї,Й,               % Cyrillic
9459         Ι,                   % Greek
9460         I,II,III},           % Roman numeral
9461 (l)I = {Ί,Ἰ,Ἱ,Ἲ,Ἳ,Ἴ,Ἵ,Ἶ,Ἷ,Ῐ,Ῑ,Ὶ,Ί}, % Greek
9462     J = {Ĵ,
9463         Ј},                  % Cyrillic
9464     K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ,Ҟ,
9465         К,Κ},                % Greek
9466     L = {Ĺ,Ļ,Ľ,Ŀ,Ł,Ḷ,Ḹ,Ḻ,Ḽ,Ƚ,Ⱡ,
9467         L},                  % Roman numeral
9468     M = {Ḿ,Ṁ,Ṃ,
9469         М,Ӎ,                 % Cyrillic
9470         Μ,                   % Greek
9471         M},                  % Roman numeral
9472     N = {Ñ,Ń,Ņ,Ň,Ŋ,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ,
9473         Ν},                  % Greek
9474     O = {Ò,Ó,Ô,Õ,Ö,Ø,Ō,Ŏ,Ő,Ơ,Ǒ,Ǫ,Ǭ,Ǿ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ,Ɵ,/Odieresis.deu,
9475         О,Ӧ,Ѳ,Ӫ,             % Cyrillic
9476         Ό,Ὀ,Ὁ,Ὂ,Ὃ,Ὄ,Ὅ,Ὸ,Ό}, % Greek
9477 % (l)O = {Ό,Ὀ,Ὁ,Ὂ,Ὃ,Ὄ,Ὅ,Ὸ,Ό}, % (accents not protruded)
9478     P = {Ṕ,Ṗ,Ҏ,
9479         Р,Ҏ,                 % Cyrillic
9480         Ρ},                  % Greek
9481 (l)P = {Ῥ},                   % Greek
9482     Q = {Ԛ},                 % Cyrillic
9483     R = {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ,Ɍ,},
9484     S = {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ,
9485         Ѕ},                  % Cyrillic
9486     T = {Ţ,Ť,Ŧ,Ț,Ṫ,Ṭ,Ṯ,Ṱ,
9487         Т,Ҭ,                 % Cyrillic
9488         Τ},                  % Greek
9489     U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ư,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự}, % /Udieresis.deu ?
9490     V = {Ṽ,Ṿ,/U.LAT,
9491         V},                  % Roman numeral
```

```
9492    W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,W̦,
9493         W},              % Cyrillic
9494    X = {X̌,Ẍ,
9495         X,X,X̣,X,         % Cyrillic
9496         X,               % Greek
9497         X},              % Roman numeral
9498    Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Y̦,Ỷ,Ỹ,
9499         Y,Y},   % Cyrillic
9500    Z = {Ź,Ż,Ž,Ẑ,Z̦,Ẓ,Ẕ,
9501         Z},              % Greek
9502    a = {à,á,â,ã,ä,ā,ă,ą,ǎ,ȧ,ȁ,ȃ,å,â,a̧,a̦,a̮,á,à,â,å,â,â,ã,â,å,ã,ǟ,â,á,
9503         a,å,ä},          % Cyrillic
9504    b = {ḃ,ḅ,b̦},
9505    c = {ç,ć,ĉ,ċ,č,c̓,
9506         c,ç,             % Cyrillic
9507         c},              % Roman numeral
9508    d = {đ,ḑ,ḋ,ḍ,ḏ,ḓ,d̦,
9509         d},              % Roman numeral
9510    e = {è,é,ê,ë,ē,ĕ,é,ę,ě,ȅ,ȇ,ȩ,ḕ,ḗ,ḙ,ḛ,ḝ,ê,ế,ề,ê,ê,
9511         e,è,ë,ĕ},        % Cyrillic
9512    f = {ḟ,ff,/f.long,/f.DEU,/f_f},
9513    fl = {ffl,/longs_l,/longs_longs_l,/f_l},
9514    fi = {ffi,/longs_i,/longs_longs_i,/f_i},
9515    /f.short = {/f_f.short},
9516    g = {ĝ,ğ,ġ,ģ,ǧ,ḡ,g̦,ǵ},
9517    h = {ĥ,ħ,ḣ,ḥ,ḧ,ḩ,ḫ,ḫ,ĥ,h̦,
9518         h,ḥ},            % Cyrillic
9519    i = {ì,í,î,ï,ĩ,ī,ĭ,į,ı,ǐ,ȉ,ȋ,ḭ,ḯ,í,/i.TRK,
9520         i,ï,             % Cyrillic
9521         i,ii,iii},       % Roman numeral
9522    j = {ĵ,ǰ,
9523         j},              % Cyrillic
9524    k = {ķ,ǩ,ḱ,ḳ,ḵ,k̦},
9525    l = {ĺ,ļ,ľ,ŀ,ł,ḷ,ḹ,ḻ,ḽ,ł,
9526         l,               % palochka
9527         l},              % Roman numeral
9528    m = {ḿ,ṁ,ṃ,
9529         m},              % Roman numeral
9530    n = {ñ,ń,ņ,ň,ŋ,ṅ,ṇ,ṉ,ṋ,ǹ},  % 'n
9531    o = {ò,ó,ô,õ,ö,ø,ō,ŏ,ő,ǒ,ǫ,ǭ,ǿ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ó,ŏ,ò,ò,o̧,ó,ô,ŏ,ô,ộ,ò,ổ,ö,ǿ,ȍ,ȏ,ő,o̦,
9532         o,ö},            % Cyrillic
9533    p = {ṕ,ṗ,
9534         p,p},            % Cyrillic
9535    q = {q},              % Cyrillic
9536    r = {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
9537    s = {ś,ŝ,ş,š,ș,ṡ,ṣ,ṥ,ṧ,ṩ,
9538         s},              % Cyrillic
9539    t = {ţ,ť,ŧ,ț,ṫ,ṭ,ṯ,ṱ,ẗ},
9540    u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ, u,ù,ú,û,ü,ũ,ư,/u.LATmedi},
9541    v = {ṽ,ṿ,
9542         v},              % Roman numeral
9543    w = {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ,
9544         w},              % Cyrillic
9545    x = {ẋ,ẍ,
9546         x,x,             % Cyrillic
9547         x},              % Roman numeral
9548    y = {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,y̦,ỷ,ỹ,
9549         y,ȳ,ÿ,ý,ỳ},      % Cyrillic
9550    z = {ź,ż,ž,z,ẑ,ẓ,ẕ},
9551    Æ = {Ǽ,Ǣ,
9552         Æ},              % Cyrillic
9553    æ = {ǣ,ǽ,
9554         æ},              % Cyrillic
9555    DZ = {DŽ},
9556    Dz = {Dž},
```

```
9557        dz  = {dž},
9558    % Smallcaps
9559        /a.sc  = {/A.sc},
9560        /ae.sc  = {/AE.sc},
9561        /d.sc  = {/D.sc},
9562        /f.sc  = {/F.sc},
9563        /g.sc  = {/G.sc},
9564        /j.sc  = {/J.sc},
9565        /l.sc  = {/L.sc},
9566        /o.sc  = {/O.sc},
9567        /oe.sc  = {/OE.sc},
9568        /q.sc  = {/Q.sc},
9569        /r.sc  = {/R.sc},
9570        /t.sc  = {/T.sc},
9571        /y.sc  = {/Y.sc},
9572     % Cyrillic
9573        Г  = {Ґ,Ғ,Ѓ,Ӷ},
9574        Ж  = {Җ,Ӂ,Ж̌,Ӝ},
9575        З  = {Ҙ,Ӟ},
9576        И  = {Й,Ѝ,Ӥ,Ҋ,Ӣ},
9577        К  = {Қ,Ҟ,Ҝ,К̨,Ќ,Ӄ},
9578        Л  = {Љ,Ӆ,Л̡},
9579        П  = {Ҧ},
9580        У  = {Ӯ,Ў,Ӱ,Ӳ},
9581        Ц  = {Ҵ,Ц̓},
9582        Ч  = {Ҷ,Ӵ,Ҹ,Ӌ},
9583        Ш  = {Щ},
9584        Ы  = {Ӹ},
9585        Ь  = {Ҍ},
9586        Э  = {Ӭ},
9587        Ѵ  = {Ѷ},
9588        Є  = {Ё},
9589        Ә  = {Ӛ},
9590        г  = {ѓ,r,ғ,ӷ},
9591        ж  = {җ,ӂ,ж̌,ӝ},
9592        з  = {ҙ,ӟ},
9593        и  = {й,ѝ,ӥ,ҋ,ӣ},
9594        к  = {ќ,қ,ӄ,ҟ},  %  ҟ,ҝ
9595        л  = {љ,ӆ,л̡},
9596        м  = {м},
9597        н  = {ң,ӈ,ӊ,ӊ},  %   ҥ
9598        п  = {ҧ},
9599        т  = {ҭ},
9600        ц  = {ҵ},
9601        ч  = {ҷ,ӵ,ҹ,ӌ},
9602        ш  = {щ},
9603        ы  = {ӹ},
9604        э  = {ӭ},
9605        ө  = {ѳ,ӫ},
9606        ѵ  = {ѷ},
9607        ү  = {ұ},
9608        є  = {ё},
9609        ә  = {ӛ},
9610     % Greek
9611        Υ  = {Ÿ,Ύ,Ϋ,Ῠ,Ῡ},
9612    (l)Υ  = {Ὺ,Ύ,Ὑ,Ὗ,Ὕ,Ὓ,Ὺ,Ύ,Ϋ},
9613    (l)Ω  = {Ὠ,Ὡ,Ὠ,Ὡ,Ὢ,Ὣ,Ὤ,Ὥ,Ὦ,Ὧ,ᾨ,ᾩ,ᾪ,ᾫ,ᾬ,ᾭ,ᾮ,ᾯ,Ω,Ω},
9614        Ω  = {Ω,Ω},  % math
9615        Δ  = {Δ},  % math
9616        Π  = {Π},  % math
9617        α  = {ά,ὰ,ᾶ,ᾱ,ᾰ,ἀ,ἁ,ἂ,ἃ,ἄ,ἅ,ἆ,ἇ,ᾳ,ᾴ,ᾲ,ᾷ,ᾀ,ᾁ,ᾂ,ᾃ,ᾄ,ᾅ,ᾆ,ᾇ},
9618        ε  = {έ,ὲ,ἐ,ἑ,ἒ,ἓ,ἔ,ἕ},
9619        η  = {ή,ὴ,ῆ,ἠ,ἡ,ἢ,ἣ,ἤ,ἥ,ἦ,ἧ,ῃ,ῄ,ῂ,ῇ,ᾐ,ᾑ,ᾒ,ᾓ,ᾔ,ᾕ,ᾖ,ᾗ},
9620        ι  = {ί,ὶ,ῖ,ῐ,ῑ,ϊ,ΐ,ῒ,ῗ,ἰ,ἱ,ἲ,ἳ,ἴ,ἵ,ἶ,ἷ},
9621        ο  = {ό,ǫ,ὸ,ό,ὀ,ὁ,ὂ,ὃ,ὄ,ὅ},
```

```
9622      ρ = {ṗ,ṕ},
9623      υ = {ṻ,ü,ú,ù,ǔ,ů,ū,ŭ,ũ,ǘ,ǚ,ừ,ù,ú,ů,ū,ũ,ṻ,ū,ũ},
9624      ω = {ǿ,ò,ó,ō,ô,ŏ,ő,ō,ǒ,ò,ó,ǫ,ǫ́,ǫ̀,ǭ,ǫ̂,ǫ̌,ǫ̆,ọ,ọ́,ọ̀,ōo,ọ̄},
9625   % other
9626      (1) = {(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15),(16),(17),(18),(19),(20)},
9627      (a) = {(b),(c),(d),(e),(f),(g),(h),(i),(j),(k),(l),(m),(n),(o),(p),(q),(r),(s),(t),(u),(v),(w),(x),(y),(z)},
9628      Ⓐ = {Ⓑ,Ⓒ,Ⓓ,Ⓔ,Ⓕ,Ⓖ,Ⓗ,Ⓘ,Ⓙ,Ⓚ,Ⓛ,Ⓜ,Ⓝ,Ⓞ,Ⓟ,Ⓠ,Ⓡ,Ⓢ,Ⓣ,Ⓤ,Ⓥ,Ⓦ,Ⓧ,Ⓨ,Ⓩ},
9629      ! = {‼},
9630      ? = {⁇},
9631      . = {/onedotleader},
9632      /endash = {/figuredash},
9633   }
```

9634 ⟨*/EBGaramond*⟩

3.1.4 Palatino

9635 ⟨**Palatino*⟩
9636 \DeclareCharacterInheritance
9637   { encoding = {TU,EU1,EU2},
9638   family = {Palatino} }

Unfortunately, I don't have a Palatino variant containing all of the following glyphs. The settings are typeset in TEX Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as '�'. To see the real settings, consult mt-Palatino.cfg.

```
9639   { A = {À,Á,Â,Ã,Ä,Ā,Ă,Ą,Ǎ,Ā́,Ā̀,Á̈,Å,Ǟ,Â̌,Ą̊,Ạ,Ạ́,Ạ̀,Ả,Ấ,Ầ,Ẫ,Ẩ,Ắ,Ằ,Ẳ,Ẵ,AA},
9640     B = {Ḃ,Ḅ,Ḇ},
9641     C = {Ç,Ć,Ĉ,Ć,Ċ,Č},
9642     D = {Ď,Ḋ,Ḍ,Ḑ,Ḓ,Ḏ},
9643     E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ê̄,Ề,Ế,Ę̄,Ę̆,Ę̌,Ẹ,Ė,Ē̈,Ế,Ề,Ễ,Ể},
9644     F = {Ḟ},
9645     G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ,Ḡ},
9646     H = {Ĥ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ},
9647     I = {Ì,Í,Î,Ĩ,Ī,Ĭ,Į,İ,Ǐ,Ỉ,Ị,Î̃,Ḭ,Ï},
9648     J = {Ĵ},
9649     K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ},
9650     L = {Ĺ,Ļ,Ľ,Ḷ,Ḻ,Ḽ,Ḹ,Ľ,Ŀ,Ł}, % Ŀ
9651     M = {Ḿ,Ṁ,Ṃ},
9652     N = {Ñ,Ń,Ņ,Ň,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ},
9653     O = {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Ǒ,Ọ,Ǫ̧,Ồ,Ȫ,Ō̈,Ȯ,Ō̇,Ő́,Ȫ,Ȱ,Ọ́,Ọ,Ọ̀,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ},
9654     P = {Ṕ,Ṗ},
9655     R = {Ŕ,Ŗ,Ř,Ȓ,Ṙ,Ṛ,Ŗ,Ṝ,Ṟ},
9656     S = {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ},
9657     T = {Ţ,Ť,Ṭ,Ṫ,Ṱ,Ṯ,Ṱ},
9658     U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ṻ,Ǖ,Ǘ,Ǚ,Ǜ,Ȗ,Ụ,Ụ̄,Ụ̀,Ứ,Ṳ,Ụ,Ủ,Ử,Ừ,Ữ,Ự},
9659     V = {Ṽ,Ṿ},
9660     W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ},
9661     X = {Ẋ,Ẍ},
9662     Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ},
9663     Z = {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ},
9664     a = {à,á,â,ã,ä,ā,ă,ą,ǎ,ā́,ā̀,á̈,å,ǟ,â̌,ą̊,ạ,ạ́,ạ̀,ả,ấ,ầ,ẫ,ẩ,ắ,ằ,ẳ,ẵ}, % aͻ
9665     b = {ḃ,ḅ,ḇ},
9666     c = {ç,ć,ĉ,ċ,č,ć̣},
9667     d = {ď,ḋ,ḍ,ḑ,ḓ,ḏ},
9668     e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ẹ,ê̄,ề,ế,ę̄,ę̆,ę̌,ẹ,ė,ē̈,ế,ề,ễ,ể},
9669     f = {ḟ,ff},
9670     g = {ĝ,ğ,ġ,ģ,ǧ,ǵ,ḡ},
9671     h = {ĥ,ȟ,ḣ,ḥ,ḧ,ḩ,ḫ,ẖ},
9672     i = {ı,ì,í,î,ĩ,ī,ĭ,į,ı̇,ǐ,ỉ,ị,î̃,ḭ,ï},
9673     j = {ĵ,ǰ},
9674     k = {ķ,ǩ,ḱ,ḳ,ḵ},
9675     l = {ĺ,ļ,ḷ,ḻ,ḽ,ḹ}, % ŀ
```

```
9676        m = {ḿ,ṁ,m̤},
9677        n = {ñ,ń,ņ,ň,ǹ,ṅ,ṇ,ṉ,ṋ},  % 'n
9678        o = {ò,ó,ô,õ,ö,ō,ŏ,ő,ǒ,ǫ,ǭ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ȫ,ȯ,ȱ,ǫ,o̧,ò,ó,ồ,ổ,õ,ớ,ờ,ở,ỡ,ợ},
9679        p = {ṕ,ṗ},
9680        r = {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
9681        s = {ś,ŝ,ş,š,ṡ,ṣ,ṥ,ṧ,ṩ},
9682        t = {ţ,ť,ṫ,ṭ,ṱ,ṭ,ẗ},  % ŧ
9683        u = {ù,ú,û,ü,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự},
9684        v = {ṽ,ṿ},
9685        w = {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ},
9686        x = {ẋ,ẍ},
9687        y = {ý,ÿ,ŷ,ȳ,ẏ,ỳ,ỵ,ỷ,ỹ},
9688        z = {ź,ż,ž,ẑ,ẓ,ẕ},
9689    }
```
9690 ⟨/*Palatino*⟩

3.1.5   Basic glyph set

There are quite a few fonts out there that don't even fill the T1 glyph set. To prevent a plethora of warnings, they may be aliased to the surrogate font TU-basic. Examples of such fonts are: Lato, Fontin and Bergamo.

9691 ⟨**TU-basic*⟩
```
9692 \DeclareCharacterInheritance
9693    { encoding = {TU,EU1,EU2},
9694      family   = {TU-basic} }
9695    { A = {À,Á,Â,Ã,Ä,Å},
9696      a = {à,á,â,ã,ä,å},
9697      C = {Ç},
9698      c = {ç},
9699      D = {Đ},
9700      E = {È,É,Ê,Ë},
9701      e = {è,é,ê,ë},
9702      I = {Ì,Í,Î,Ï},
9703      i = {ì,í,î,ï,ı},
9704      L = {Ł},
9705      l = {ł},
9706      N = {Ñ},
9707      n = {ñ},
9708      O = {Ø,Ò,Ó,Ô,Õ,Ö},
9709      o = {ø,ò,ó,ô,õ,ö},
9710      S = {Š},
9711      s = {š},
9712      U = {Ù,Ú,Û,Ü},
9713      u = {ù,ú,û,ü},
9714      Y = {Ý,Ÿ},
```

For some reason, the ÿ in the next line comes out as ß. Don't worry, there's really a `y diaeresis` in the source.

```
9715      y = {ý,ß},
9716      Z = {Ž},
9717      z = {ž}
9718    }
```
9719 ⟨/*TU-basic*⟩

3.1.6   Empty glyph set

Other fonts, e.g., the self-professedly awesone Font Awesome font, have no meaningful glyph arsenal at all, and should therefore be aliased so that empty settings are applied.

9720 ⟨**TU-empty*⟩
9721 \DeclareCharacterInheritance

```
9722   { encoding = {TU,EU1,EU2},
9723     family   = {TU-empty} }
9724   { }
9725 ⟨/TU-empty⟩
```

3.2 Character protrusion

```
9726
9727 %%% --------------------------------------------------------------------
9728 %%% PROTRUSION
9729
```

3.2.1 Latin Modern Roman/New Computer Modern

```
9730 ⟨*LatinModernRoman|NewComputerModern⟩
9731 \SetProtrusion
9732 ⟨LatinModernRoman⟩  [ name     = LMR-default ]
9733 ⟨NewComputerModern⟩  [ name     = NCM-default ]
9734 ⟨LatinModernRoman⟩  { encoding = {TU,EU1,EU2},
9735 ⟨LatinModernRoman⟩    family   = Latin Modern Roman }
9736 ⟨NewComputerModern⟩  { }
9737   {
9738     A = {50,50},
9739     Æ = {50,  },
9740     F = {  ,50},
9741     J = {50,  },
9742     K = {  ,50},
9743     L = {  ,50},
9744     T = {50,50},
9745     V = {50,50},
9746     W = {50,50},
9747     X = {50,50},
9748     Y = {50,50},
9749     k = {  ,50},
9750     r = {  ,50},
9751     t = {  ,70},
9752     v = {50,50},
9753     w = {50,50},
9754     x = {50,50},
9755     y = {50,70},
9756     0 = {  ,50},
9757     1 = {100,200},
9758     2 = {50,50},
9759     3 = {50,50},
9760     4 = {70,70},
9761     5 = {  ,50},
9762     6 = {  ,50},
9763     7 = {50,100},
9764     8 = {  ,50},
9765     9 = {  ,50},
9766     . = { ,700},
9767   {,}= { ,500},
9768     : = { ,500},
9769     ; = { ,500},
9770     ! = { ,100},
9771     ? = { ,200},
9772     @ = {50,50},
9773     ~ = {200,250},
9774    \% = {50,50},
9775     * = {300,300},
9776     + = {250,250},
9777     - = {400,500},  % /hyphen
9778     – = {400,300},  % /endash
9779     — = {300,200},  % /emdash
9780     _ = {200,200},  % /underscore
```

```
9781      / = {200,300},
9782      /backslash     = {200,300},
9783      ' = {300,400}, % /quotesingle
9784      ' = {300,400},  ' = {300,400},
9785      " = {300,300},  " = {300,300},
9786      , = {400,400},  „ = {400,400},
9787      ‹ = {400,400},  › = {300,500},
9788      « = {300,200},  » = {100,400},
9789      ¡ = {100,   },  ¿ = {100,   },
9790      ( = {300,   },  ) = {   ,300},
9791      < = {200,100}, > = {100,200},
9792      /braceleft = {400,200}, /braceright = {200,400},
9793      /angleleft = {400,   }, /angleright = {   ,400},
9794      † = {100,100},
9795      ‡ = { 80, 80},
9796      • = {200,200},
9797      · = {400,450}, % / periodcentered
9798      ℃ = { 80, 50},
9799      ℂ = {   , 50},
9800      ° = {400,400},
9801      ™ = {100,200},
9802      © = {100,100},
9803      ® = {100,100},
9804      ª = {100,200},
9805      º = {100,200},
9806      ¹ = {200,250},
9807      ² = { 50,100},
9808      ³ = { 50,100},
9809      ¬ = {200,   },
9810      — = {300,300},
9811      ± = {150,200},
9812      × = {150,250},
9813      ÷ = {150,250},
9814      € = {100,   },
9815  ⟨*LatinModernRoman⟩
9816      /one.oldstyle   = {100,100},
9817      /two.oldstyle   = { 50, 50},
9818      /three.oldstyle = { 30, 80},
9819      /four.oldstyle  = { 50, 50},
9820      /seven.oldstyle = { 50, 80},
9821  ⟨/LatinModernRoman⟩
9822  ⟨*NewComputerModern⟩
9823      A = {50,50},   % /Alphatonos
9824      Ἁ = {120,50},  %
9825      Ἀ = {120,50},  %
9826      Ά = {80,50},   %
9827      Ἇ = {220,50},  %
9828      Ἆ = {220,50},  %
9829      Ἅ = {170,50},  %
9830      Ἄ = {170,50},  %
9831      Ἃ = {190,50},  %
9832      Ἂ = {190,50},  %
9833      Ἁ = {150,50},  %
9834      Ά = {80,50},   %
9835      Ἇ = {220,50},  %
9836      Ἆ = {220,50},  %
9837      Ἅ = {170,50},  %
9838      Ἄ = {170,50},  %
9839      Ἃ = {210,50},  %
9840      Ἂ = {210,50},  %
9841      /uni1FBC.alt = {,205}, % Alpha prosgegrammeni
9842      /uni1F88.alt = {50,190}, %Alpha psili prosgegrammeni
9843      /uni1F89.alt = {,200}, %Alpha dasia prosgegrammeni
9844      /uni1F8A.alt = {130,180}, %Alpha psili baria prosgegrammeni
9845      /uni1F8B.alt = {130,190}, %Alpha dasia baria prosgegrammeni
```

```
9846      /uni1F8C.alt = {100,190}, %Alpha psili oxia prosgegrammeni
9847      /uni1F8D.alt = {70,190}, %Alpha dasia oxia prosgegrammeni
9848      /uni1F8E.alt = {120,190}, %Alpha psili perispomeni prosgegrammeni
9849      /uni1F8F.alt = {120,190}, %Alpha dasia perispomeni prosgegrammeni
9850    %
9851      /uni1FCC.alt = {,205}, % Eta prosgegrammeni
9852      /uni1F98.alt = {185,170}, %Eta psili prosgegrammeni
9853      /uni1F99.alt = {185,170}, %Eta dasia prosgegrammeni
9854      /uni1F9A.alt = {220,170}, %Eta psili baria prosgegrammeni
9855      /uni1F9B.alt = {220,170}, %Eta dasia baria prosgegrammeni
9856      /uni1F9C.alt = {220,170}, %Eta psili oxia prosgegrammeni
9857      /uni1F9D.alt = {220,170}, %Eta dasia oxia prosgegrammeni
9858      /uni1F9E.alt = {255,170}, %Eta psili perispomeni prosgegrammeni
9859      /uni1F9F.alt = {255,170}, %Eta dasia perispomeni prosgegrammeni
9860    %
9861      O = {95,50}, %
9862 ⟨/NewComputerModern⟩
9863      Γ = {   ,180}, % /Gamma
9864 ⟨LatinModernRoman⟩    Δ = {100,100}, % /Delta
9865 ⟨NewComputerModern⟩    Δ = {50,50}, % /Delta
9866      Θ = { 50, 50}, % /Theta
9867 ⟨LatinModernRoman⟩    Λ = {100,100}, % /Lambda
9868 ⟨NewComputerModern⟩    Λ = {50,50}, % /Lambda
9869 %    Ξ = {,},      % /Xi
9870 %    Π = {,},      % /Pi
9871      Σ = { 50, 50}, % /Sigma
9872 ⟨LatinModernRoman⟩    Υ = {100,100}, % /Upsilon
9873 ⟨NewComputerModern⟩    Υ = {80,80}, % /Upsilon
9874      Φ = { 50, 50}, % /Phi
9875      Ψ = { 50, 50}, % /Psi
9876 ⟨*NewComputerModern⟩
9877      Ω = { 20, 30}, % /Omega
9878      Ὠ = {150,30},
9879      Ὀ = {220,30},
9880      Ὡ = {205,30},
9881      Ὢ = {285,30},
9882      Ὣ = {285,30},
9883      Ὤ = {270,30},
9884      Ὥ = {270,30},
9885      Ὦ = {310,30},
9886      Ὧ = {310,30},
9887      Ὠ = {205,30},
9888      Ὡ = {205,30},
9889      Ὢ = {285,30},
9890      Ὣ = {285,30},
9891      Ὤ = {270,30},
9892      Ὥ = {270,30},
9893      Ὦ = {310,30},
9894      Ὧ = {310,30},
9895      /uni1FFC.alt = {,230}, % Omega prosgegrammeni
9896      /uni1FA8.alt = {185,190}, %Omega psili prosgegrammeni
9897      /uni1FA9.alt = {185,190}, %Omega dasia prosgegrammeni
9898      /uni1FAA.alt = {220,190}, %Omega psili baria prosgegrammeni
9899      /uni1FAB.alt = {220,190}, %Omega dasia baria prosgegrammeni
9900      /uni1FAC.alt = {220,190}, %Omega psili oxia prosgegrammeni
9901      /uni1FAD.alt = {220,190}, %Omega dasia oxia prosgegrammeni
9902      /uni1FAE.alt = {255,190}, %Omega psili perispomeni prosgegrammeni
9903      /uni1FAF.alt = {255,190}, %Omega dasia perispomeni prosgegrammeni
9904    %
9905      α = {,50},
9906      γ = {50,50},
9907      ζ = {,50},
9908      ϑ = {30,40},
9909      ι = {,50},
9910      ῒ = {-20,-30},
```

```
9911        ϰ = {50,50},
9912        λ = {50,50},
9913        ν = {50,25},
9914        π = {50,50},
9915        σ = {,50},
9916        ς = {,50},
9917        τ = {50,50},
9918        χ = {50,50},
9919        ψ = {50,50},
9920 %    /uni1F98.alt = {,},
```

CMU Serif doesn't include ∗.end glyphs, and the OldStyle numbers' names differ.

```
9921      }
9922
9923 \SetProtrusion
9924    [ name     = NCM-TU,
9925      load     = NCM-default ]
9926    { encoding = {TU,EU1,EU2},
9927      family   = {New Computer Modern} }
9928    {
9929      /a.end = {,330},
9930      /e.end = {,350},
9931      /k.alt = { ,50},
9932      /r.end = {,300},
9933      /m.end = {,200},
9934      /n.end = {,300},
9935      /one.oldstyle   = {100,100},
9936      /two.oldstyle   = { 50, 50},
9937      /three.oldstyle = { 30, 80},
9938      /four.oldstyle  = { 50, 50},
9939      /seven.oldstyle = { 50, 80},
9940    }
9941
9942 \SetProtrusion
9943    [ name     = CMU-TU,
9944      load     = NCM-default ]
9945    { encoding = {TU,EU1,EU2},
9946      family   = {CMU Serif} }
9947    {
9948      /oneoldstyle   = {100,100},
9949      /twooldstyle   = { 50, 50},
9950      /threeoldstyle = { 30, 80},
9951      /fouroldstyle  = { 50, 50},
9952      /sevenoldstyle = { 50, 80},
9953 ⟨/NewComputerModern⟩
9954    }
9955
9956 \SetProtrusion
9957 ⟨LatinModernRoman⟩    [ name     = LMR-it ]
9958 ⟨NewComputerModern⟩   [ name      = NCM-it ]
9959 ⟨LatinModernRoman⟩    { encoding = {TU,EU1,EU2},
9960 ⟨LatinModernRoman⟩      family   = Latin Modern Roman,
9961 ⟨LatinModernRoman⟩      shape    = {it,sl}        }
9962 ⟨NewComputerModern⟩   { }
9963    {
9964      A = {125,100},
9965      Æ = {125,-55},
9966      B = {90,-40},
9967      C = {145,-75},
9968      D = {75, -28},
9969      E = {80,-55},
9970      F = {85,-80},
9971      G = {153,-15},
9972      H = {73,-60},
9973      I = {140,-120},
```

```
9974        IJ = {140,-80},
9975        J = {135,-80},
9976        K = {70,-30},
9977        L = {87, 40},
9978        M = {67,-45},
9979        N = {75,-55},
9980        O = {150,-30},
9981        Œ = {150,-55},
9982        P = {82,-50},
9983        Q = {150,-30},
9984        R = {75, 15},
9985        S = {90,-65},
9986        $ = {100,-20},
9987        T = {220,-85},
9988        U = {230,-55},
9989        V = {260,-60},
9990        W = {185,-55},
9991        X = {70,-30},
9992        Y = {250,-60},
9993        Z = {90,-60},
9994        a = {150,-10},
9995        b = {170,    },
9996        c = {173,-10},
9997        d = {150,-55},
9998        e = {180, },
9999        f = {   ,-250},
10000       g = {150,-10},
10001       h = {100, },
10002       i = {210, },
10003       ij = {210,-40},
10004       j = {   ,-40},
10005       k = {110,-50},
10006       l = {240,-110},
10007       m = {80, },
10008       n = {115, },
10009       o = {155, },
10010       q = {170,-40},
10011       r = {155,-40},
10012       s = {130, },
10013       t = {230,-10},
10014       u = {120, },
10015       v = {140,-25},
10016       w = {98,-20},
10017       x = {65,-40},
10018       y = {130,-20},
10019       z = {110,-80},
10020       0 = {170,-85},
10021       1 = {230,110},
10022       2 = {130,-70},
10023       3 = {140,-70},
10024       4 = {130,80},
10025       5 = {160, },
10026       6 = {175,-30},
10027       7 = {250,-150},
10028       8 = {130,-40},
10029       9 = {155,-80},
10030       . = { ,500},
10031       {,}= { ,450},
10032       : = { ,300},
10033       ; = { ,300},
10034       & = {130,30},
10035       \% = {180,50},
10036       * = {380,20},
10037       + = {180,200},
10038       @ = {180,10},
```

```
10039      ~ = {200,150},
10040      ( = {300, },     ) = {  ,70},
10041      / = {100,100},
10042      - = {500,300}, % /hyphen
10043      – = {500,300}, % /endash
10044      — = {400,170}, % /emdash
10045      _ = {100,200}, % /underscore
10046      ' = {300,400}, % /quotesingle
10047      " = {500,300},
10048      ' = {800,200},   ' = {800,-20},
10049      " = {540,100},   " = {500,100},
10050      , = {300,700},   „ = {200,600},
10051      ‹ = {500,300},   › = {400,400},
10052      « = {400,100},   » = {200,300},
10053      ¡ = {200,   },   ¿ = {200,   },
10054      < = {300,100},   > = {200,100},
10055      /backslash   = {300,300},
10056      /braceleft   = {400,100},   /braceright    = {200,200},
10057      † = {200, 80},
10058      ‡ = {120, 80},
10059      • = {220,100},
10060      · = {550,300}, % / periodcentered
10061      ℃ = {170,   },
10062      ℂ = {100, 50},
10063      ¶ = {200,   },
10064      ° = {500,300},
10065      ™ = {200, 70},
10066      © = { 50, 70},
10067      ® = { 50, 70},
10068      ª = {140,100},
10069      º = {140,100},
10070      ¹ = {400,150},
10071      ² = {250, 80},
10072      ³ = {250, 80},
10073      ¬ = {250, 80},
10074      − = {300,200},
10075      ± = {150,170},
10076      × = {200,200},
10077      ÷ = {200,200},
10078      € = {150,   },
10079 ⟨*LatinModernRoman⟩
10080      /one.oldstyle   = {100,100},
10081      /two.oldstyle   = {100, 80},
10082      /three.oldstyle = { 80, 50},
10083      /four.oldstyle  = { 80, 80},
10084      /five.oldstyle  = { 50,   },
10085      /six.oldstyle   = { 50,   },
10086      /seven.oldstyle = { 80, 80},
10087      /eight.oldstyle = { 50,   },
10088 ⟨/LatinModernRoman⟩
10089      Γ = {100,120}, % /Gamma
10090      Δ = {120,100}, % /Delta
10091      Θ = {120, 50}, % /Theta
10092 ⟨LatinModernRoman⟩     Λ = {130,100}, % /Lambda
10093 ⟨NewComputerModern⟩    Λ = {160,100}, % /Lambda
10094      Ξ = {100,},    % /Xi
10095      Π = {100,},    % /Pi
10096      Σ = {100, 50}, % /Sigma
10097 ⟨LatinModernRoman⟩     Υ = {180,100}, % /Upsilon
10098 ⟨NewComputerModern⟩    Υ = {260,100}, % /Upsilon
10099      Φ = {130, 70}, % /Phi
10100      Ψ = {130, 50}, % /Psi
10101      Ω = { 50,},    % /Omega
10102 ⟨*NewComputerModern⟩
10103      Ὰ = {190,50},  %
```

```
10104    Ἀ = {220,50},  %
10105    Ἁ = {200,50},  %
10106    Ἂ = {300,50},  %
10107    Ἃ = {300,50},  %
10108    Ἄ = {300,50},  %
10109    Ἅ = {300,50},  %
10110    Ἆ = {320,50},  %
10111    Ἇ = {320,50},  %
10112    Ᾰ = {200,50},  %
10113    Ᾱ = {200,50},  %
10114    Ὰ = {300,50},  %
10115    Ά = {300,50},  %
10116    ΑΙ = {300,50},  %
10117    ΆΙ = {300,50},  %
10118    Α͂ = {320,50},  %
10119    Α͂Ι = {320,50},  %
10120    /uni1FBC.alt = {,205}, % Alpha prosgegrammeni
10121    /uni1F88.alt = {50,190}, %Alpha psili prosgegrammeni
10122    /uni1F89.alt = {,200}, %Alpha dasia prosgegrammeni
10123    /uni1F8A.alt = {130,180}, %Alpha psili baria prosgegrammeni
10124    /uni1F8B.alt = {130,190}, %Alpha dasia baria prosgegrammeni
10125    /uni1F8C.alt = {100,190}, %Alpha psili oxia prosgegrammeni
10126    /uni1F8D.alt = {70,190}, %Alpha dasia oxia prosgegrammeni
10127    /uni1F8E.alt = {120,190}, %Alpha psili perispomeni prosgegrammeni
10128    /uni1F8F.alt = {120,190}, %Alpha dasia perispomeni prosgegrammeni
10129    %
10130    /uni1FCC.alt = {,205}, % Eta prosgegrammeni
10131    /uni1F98.alt = {185,170}, %Eta psili prosgegrammeni
10132    /uni1F99.alt = {185,170}, %Eta dasia prosgegrammeni
10133    /uni1F9A.alt = {220,170}, %Eta psili baria prosgegrammeni
10134    /uni1F9B.alt = {220,170}, %Eta dasia baria prosgegrammeni
10135    /uni1F9C.alt = {220,170}, %Eta psili oxia prosgegrammeni
10136    /uni1F9D.alt = {220,170}, %Eta dasia oxia prosgegrammeni
10137    /uni1F9E.alt = {255,170}, %Eta psili perispomeni prosgegrammeni
10138    /uni1F9F.alt = {255,170}, %Eta dasia perispomeni prosgegrammeni
10139    %
10140    Ο = {95,50}, %
10141    Ω = {120, 30}, % /Omega
10142    Ω = {160,30},
10143    Ὠ = {250,30},
10144    Ὡ = {250,30},
10145    Ὢ = {300,30},
10146    Ὣ = {300,30},
10147    Ὤ = {300,30},
10148    Ὥ = {300,30},
10149    Ὦ = {330,30},
10150    Ὧ = {330,30},
10151    Ω = {30,30},
10152    Ὠ = {230,30},
10153    Ὡ = {230,30},
10154    Ὢ = {300,30},
10155    Ὣ = {300,30},
10156    Ὤ = {300,30},
10157    Ὥ = {300,30},
10158    Ὦ = {330,30},
10159    Ὧ = {330,30},
10160    /uni1FFC.alt = {,230}, % Omega prosgegrammeni
10161    /uni1FA8.alt = {185,190}, %Omega psili prosgegrammeni
10162    /uni1FA9.alt = {185,190}, %Omega dasia prosgegrammeni
10163    /uni1FAA.alt = {220,190}, %Omega psili baria prosgegrammeni
10164    /uni1FAB.alt = {220,190}, %Omega dasia baria prosgegrammeni
10165    /uni1FAC.alt = {220,190}, %Omega psili oxia prosgegrammeni
10166    /uni1FAD.alt = {220,190}, %Omega dasia oxia prosgegrammeni
10167    /uni1FAE.alt = {255,190}, %Omega psili perispomeni prosgegrammeni
10168    /uni1FAF.alt = {255,190}, %Omega dasia perispomeni prosgegrammeni
```

```
10169    %
10170      α = {50,50},
10171      γ = {100,50},
10172      δ = {30,50},
10173      ε = {30,},
10174      ζ = {20,50},
10175      ϑ = {30,40},
10176      ι = {,50},
10177      ϊ = {-20,-30},
10178      ϰ = {50,50},
10179      λ = {-20,50},
10180      ν = {50,25},
10181      o ={40,},
10182      π = {50,50},
10183      σ = {40,50},
10184      ς = {20,50},
10185      τ = {50,50},
10186      υ = {80,},
10187      φ = {80,},
10188      χ = {20,},
10189      ψ = {80,},
10190 %    /uni1F98.alt = {,},
10191      }
10192
10193 \SetProtrusion
10194    [ name      = NCM-it-TU,
10195      load      = NCM-it ]
10196    { encoding = {TU,EU1,EU2},
10197      family    = {New Computer Modern},
10198      shape     = {it,sl} }
10199    {
10200     /a.end = {,330}, %Fix
10201     /e.end = {,350}, %Fix
10202     /k.alt = {   ,50}, %Fix
10203     /r.end = {,300}, %Fix
10204     /m.end = {,200}, %Fix
10205     /n.end = {,300}, %Fix
10206     /one.oldstyle   = {100,100},
10207     /two.oldstyle   = {100, 80},
10208     /three.oldstyle = { 80, 50},
10209     /four.oldstyle  = { 80, 80},
10210     /five.oldstyle  = { 50,   },
10211     /six.oldstyle   = { 50,   },
10212     /seven.oldstyle = { 80, 80},
10213     /eight.oldstyle = { 50,   },
10214    }
10215
10216 \SetProtrusion
10217    [ name      = CMU-it-TU,
10218      load      = NCM-it ]
10219    { encoding = {TU,EU1,EU2},
10220      family    = {CMU Serif},
10221      shape     = {it,sl} }
10222    {
10223     /oneoldstyle   = {100,100},
10224     /twooldstyle   = {100, 80},
10225     /threeoldstyle = { 80, 50},
10226     /fouroldstyle  = { 80, 80},
10227     /fiveoldstyle  = { 50,   },
10228     /sixoldstyle   = { 50,   },
10229     /sevenoldstyle = { 80, 80},
10230     /eightoldstyle = { 50,   },
10231 ⟨/NewComputerModern⟩
10232    }
10233 ⟨/LatinModernRoman|NewComputerModern⟩
```

3.2.2 Charis SIL

```
10234 ⟨*CharisSIL⟩
10235 \SetProtrusion
10236   [ name      = Charis-default ]
10237   { encoding = {TU,EU1,EU2},
10238     family    = Charis SIL }
10239   {
10240       A  =  {50,50},
10241       Æ  =  {50,50},
10242       C  =  {50,   },
10243       D  =  {  ,50},
10244       F  =  {  ,50},
10245       G  =  {50,   },
10246       J  =  {100, },
10247       K  =  {  ,50},
10248       L  =  {  ,50},
10249       Ŀ  =  {  ,100},
10250       O  =  {50,50},
10251       Œ  =  {50,   },
10252       P  =  {  ,50},
10253       Q  =  {50,70},
10254       R  =  {  ,50},
10255       ß  =  {  ,40}, % capital sharp s
10256       T  =  {50,50},
10257       V  =  {50,50},
10258       W  =  {50,50},
10259       X  =  {50,50},
10260       Y  =  {50,50},
10261       k  =  {  ,50},
10262       ŀ  =  {  ,150},
10263       r  =  {  ,50},
10264       t  =  {  ,50},
10265       v  =  {50,50},
10266       w  =  {50,50},
10267       x  =  {50,50},
10268       y  =  {  ,50},
10269       1  =  {150,150},
10270       2  =  {50,50},
10271       3  =  {50,   },
10272       4  =  {100,50},
10273       6  =  {50,   },
10274       7  =  {50,80},
10275       9  =  {50,50},
10276       .  =  {  ,600},
10277     {,}=  {  ,500},
10278       :  =  {  ,400},
10279       ;  =  {  ,300},
10280       !  =  {  ,100},
10281       ?  =  {  ,200},
10282       @  =  {50,50},
10283       ~  =  {200,250},
10284     \%  =  {  ,50},
10285       *  =  {300,300},
10286       +  =  {200,250},
10287       /  =  {  ,200},
10288     /backslash  =  {150,200},
10289       |  =  {200,200},
10290       -  =  {400,500}, % hyphen
10291       –  =  {200,300}, % endash
10292       —  =  {150,250}, % emdash
10293       ―  =  {200,200}, % Horizontal Bar  =  \texttwelveudash
10294       ‒  =  {150,150}, % Figure Dash  =  \textthreequartersemdash
10295       _  =  {100,100},
10296     {=}  =  {100,100},
```

```
10297        ' = {300,400},   ' = {300,400},
10298        " = {300,300},   " = {300,300},
10299        , = {400,400},   „ = {300,300},
10300        ‹ = {400,300},   › = {300,400},
10301        « = {200,200},   » = {150,300},
10302        ¡ = {100,    },   ¿ = {100,    },
10303        ( = {200,    },   ) = {    ,200},
10304        < = {200,150},   > = {100,200},
10305        [ = {100,    },   ] = {    ,100},
10306      /braceleft = {200,    }, /braceright = {    ,300},
10307        † = { 80, 80},
10308        ‡ = {100,100},
10309        • = {200,200},
10310        ° = {150,200},
10311        ™ = {150,150},
10312        ¢ = { 50,    },
10313        £ = { 50,    },
10314        ¦ = {200,200},
10315        © = {100,100},
10316        ® = {100,100},
10317        ª = {100,200},
10318        º = {200,200},
10319        ¬ = {200, 50},
10320        µ = {    ,100},
10321        ¶ = {    ,100},
10322        · = {300,400},
10323        ¹ = {200,300},
10324        ² = {100,200},
10325        ³ = {100,200},
10326        € = {100,    },
10327        ± = {150,200},
10328        × = {200,200},
10329        ÷ = {250,250},
10330      /minus = {200,200},
10331        − = {200,200},
10332      % Cyrillic
10333        Б = {    ,50},
10334        Г = {    ,130},
10335        Ж = {50,50},
10336        З = {30,50},
10337        Л = {50,    },
10338        У = {50,50},
10339        Ф = {50,50},
10340        Ч = {100,    },
10341        Ъ = {    ,50},
10342        Ь = {    ,50},
10343        Э = {50,50},
10344        Ю = {    ,40},
10345        Я = {50,    },
10346        Ѵ = {50,50},
10347        Є = {50,    },
10348        Ћ = {50,100},
10349        Є = {50,    },
10350        Љ = {50,50},
10351        Њ = {    ,50},
10352        Ђ = {50,50},
10353        Ꙃ = {100,100},
10354        Ꙁ = {50,50},
10355        Ѣ = {    ,50},
10356        Ь = {    ,50},
10357        Љ = {50,80},
10358        Њ = {    ,80},
10359        Ꚋ = {50,50},
10360        Ј = {50,    },
10361        Јх = {50,40},
```

```
10362      Ҡ  =  {    ,50},
10363      Ѥ  =  {50,   },
10364      Љ  =  {    ,50},
10365      Њ  =  {    ,50},
10366      ꚉ  =  {    ,100},
10367      б  =  {50,50},
10368      г  =  {    ,70},
10369      к  =  {    ,50},
10370      л  =  {50,   },
10371      т  =  {50,50},
10372      ф  =  {50,50},
10373      ч  =  {50,   },
10374      ъ  =  {    ,50},
10375      ь  =  {    ,50},
10376      э  =  {    ,50},
10377      я  =  {50,   },
10378      љ  =  {50,   },
10379      њ  =  {    ,50},
10380      ѣ  =  {    ,50},
10381      ѵ  =  {50,50},
10382      е  =  {50,   },
10383      ь  =  {    ,50},
10384      ү  =  {50,50},
10385      ҕ  =  {    ,50},
10386      ꙥ  =  {    ,50},
10387      ꚉ  =  {    ,100},
10388      ꙋ  =  {100,100},
10389      ꙁ  =  {50,50},
10390      љ  =  {50,70},
10391      њ  =  {    ,70},
10392      ӕ  =  {50,30},
10393      љ  =  {    ,50},
10394      њ  =  {    ,50},
10395      %        Д  П  Ц  Ш  Щ  Ы  Ҕ  Њ  ꚗ  Ә  Ц  Ц  Ӟ  З  Є  ꙟ
10396      %        в  д  ж  з  и  м  н  п  ц  ш  ы  ю  ꙡ  є  ћ  ꙣ  ə  ε  ꙍ  ц  ʒ  d  ꙇ  ъ  л  ꙗ  ҏ
10397   % Greek
10398      Δ  =  {50,50},
10399      Ψ  =  {50,50},
10400      γ  =  {70,70},
10401      λ  =  {40,70},
10402      π  =  {40,50},
10403      ρ  =  {    ,50},
10404      σ  =  {    ,50},
10405      χ  =  {50,50},
10406 }
10407
10408 \SetProtrusion
10409   [ name     = Charis-it  ]
10410   { encoding = {TU,EU1,EU2},
10411     family   = Charis SIL,
10412     shape    = {it,sl}  }
10413   {
10414      C  =  {50,   },
10415      G  =  {50,   },
10416      J  =  {50,   },
10417      L  =  {50,50},
10418      O  =  {50,   },
10419      Œ  =  {50,   },
10420      Q  =  {50,   },
10421      S  =  {50,   },
10422      $  =  {50,   },
10423      T  =  {70,   },
10424      o  =  {50,50},
10425      p  =  {    ,50},
10426      q  =  {50,   },
```

```
10427        t  =  {   ,50},
10428        w  =  {   ,50},
10429        y  =  {   ,50},
10430        1  =  {150,100},
10431        3  =  {50,  },
10432        4  =  {100, },
10433        6  =  {50,  },
10434        7  =  {100, },
10435        .  =  {  ,700},
10436     {,}=  {  ,600},
10437        :  =  {  ,400},
10438        ;  =  {  ,400},
10439        ?  =  {  ,150},
10440        &  =  {  ,80},
10441       \%  =  {50,50},
10442        *  =  {300,200},
10443        +  =  {250,250},
10444        @  =  {80,50},
10445        ~  =  {150,150},
10446        /  =  {  ,150},
10447    /backslash  =  {150,150},
10448        -  =  {300,400},  % hyphen
10449        –  =  {200,300},  % endash
10450        —  =  {150,200},  % emdash
10451        _  =  {   ,100},
10452     {=}  =  {200,200},
10453        ±  =  {150,200},
10454        ×  =  {250,250},
10455        ÷  =  {250,250},
10456        °  =  {150,200},
10457        ·  =  {300,400},
10458        '  =  {400,200},    ' = {400,200},
10459        "  =  {300,200},    " = {400,200},
10460        ,  =  {200,500},    „ = {150,500},
10461        ‹  =  {300,400},    › = {200,500},
10462        «  =  {200,300},    » = {150,400},
10463        (  =  {200,   },  )  = {    ,200},
10464        <  =  {200,200},  >  = {200,200},
10465    /braceleft  =  {300,   },  /braceright = {    ,200},
10466   % Cyrillic
10467        Ж  =  {50,30},
10468        Л  =  {50,  },
10469        У  =  {50,30},
10470        Ф  =  {50,  },
10471        Ч  =  {100, },
10472        Ъ  =  {  ,50},
10473        Ь  =  {  ,50},
10474        Э  =  {50,50},
10475        Я  =  {50,  },
10476        Ѵ  =  {50,50},
10477        Љ  =  {50,50},
10478        Ꙋ  =  {140,100},
10479        Ҙ  =  {70,50},
10480        Љ  =  {50,80},
10481        Њ  =  {  ,80},
10482        Ћ  =  {50,50},
10483        г  =  {50,50},
10484        д  =  {50,30},
10485        м  =  {50,  },
10486        ф  =  {50,  },
10487        ч  =  {50,  },
10488        ъ  =  {  ,50},
10489        ь  =  {  ,50},
10490        э  =  {  ,50},
10491        я  =  {50,  },
```

```
10492        Љ  =  {50,50},
10493        Њ  =  {  ,50},
10494        Ѵ  =  {50,50},
10495        ѣ  =  {  ,50},
10496        ꙋ  =  {140,100},
10497        ꙃ  =  {70,50},
10498        Ꙛ  =  {50,70},
10499        Ꙗ  =  {  ,70},
10500     % Greek
10501        Γ  =  {  ,130},
10502        Δ  =  {50,50},
10503        Ψ  =  {50,50},
10504        γ  =  {70,70},
10505        λ  =  {40,70},
10506        π  =  {40,50},
10507        ρ  =  {  ,50},
10508        σ  =  {  ,50},
10509        χ  =  {50,50},
10510     }
```

The small caps glyph names in Charis SIL have changed with version 5.0 of the font. We try to get the names right both with LuaTEX (where we can simply query the font version) and with XƎTEX (where we check for glyph name).

```
10511
10512 % quick and dirty -- maybe we'll promote this to a
10513 % regular key some time
10514 \define@key{MT@pr@c}{command}{\csname #1\endcsname}
10515
10516 % glyph names have changed with version 5.0 of Charis SIL:
10517 % before: /a.SC, /b.SC, ...
10518 % after:  /a.sc, /b.sc, ...
10519 \ifx\MT@lua\@undefined
10520   \gdef\MT@get@CHARIS@SC{
10521     % test whether glyph "a.sc" exists
10522     \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
10523       \gdef\MT@CHARIS@SC{sc}%
10524     \else
10525       \gdef\MT@CHARIS@SC{SC}%
10526     \fi
10527   }
10528 \else
10529   \gdef\MT@get@CHARIS@SC{
10530     \gdef\MT@CHARIS@SC{\MT@lua{
10531       % check font version
10532 % --  why doesn't this work?:
10533 %    f = font.getfont(font.current());
10534 %    i = fontloader.info(f.filename);
10535 %    if (tonumber(i.version) < 5) then;
10536      if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
10537        tex.print("SC");
10538      else;
10539        tex.print("sc");
10540      end
10541   }}
10542   }
10543 \fi
10544
10545 \SetProtrusion
10546   [ name    = Charis-sc,
10547     load    = Charis-default,
10548     command = {MT@get@CHARIS@SC} ]
10549   { encoding = {TU,EU1,EU2},
10550     family   = Charis SIL,
10551     shape    = {sc}   }
```

```
10552      {
10553 %      A = {100,100}, % etc., doesn't work with \textsc
10554       /a.\MT@CHARIS@SC = {100,100},
10555       /c.\MT@CHARIS@SC = {50,  },
10556       /d.\MT@CHARIS@SC = {  ,50},
10557       /f.\MT@CHARIS@SC = {  ,50},
10558       /g.\MT@CHARIS@SC = {50,  },
10559       /j.\MT@CHARIS@SC = {100, },
10560       /k.\MT@CHARIS@SC = {  ,50},
10561       /l.\MT@CHARIS@SC = {  ,50},
10562     /f_l.\MT@CHARIS@SC = {  ,50},
10563       /o.\MT@CHARIS@SC = {50,50},
10564      /oe.\MT@CHARIS@SC = {50,  },
10565       /q.\MT@CHARIS@SC = {50,70},
10566       /r.\MT@CHARIS@SC = {  ,50},
10567       /t.\MT@CHARIS@SC = {50,100},
10568       /v.\MT@CHARIS@SC = {50,50},
10569       /w.\MT@CHARIS@SC = {50,50},
10570       /x.\MT@CHARIS@SC = {50,50},
10571       /y.\MT@CHARIS@SC = {50,50}
10572      }
10573 ⟨/CharisSIL⟩
```

3.2.3   EB Garamond

```
10574 ⟨*EBGaramond⟩
10575 \SetProtrusion
10576  [ name    = EBGaramond-TU,
10577    load    = EBGaramond-T1-LF ]
10578  { encoding = {TU,EU1,EU2},
10579    family  = EBGaramond }
10580    {
10581   /one.tosf   = {150,150},
10582   /two.tosf   = {50,50},
10583   /three.tosf = {50,50},
10584   /four.tosf  = {50,50},
10585   /five.tosf  = {50,50},
10586   /six.tosf   = {50,50},
10587   /seven.tosf = {50,80},
10588   /eight.tosf = {50,50},
10589   /nine.tosf  = {50,50},
10590   /one.lf     = {50,50},
10591   /two.lf     = {50,50},
10592   /four.lf    = {50,50},
10593   /seven.lf   = {50,50},
10594   /one.osf    = {50,50},
10595   /two.osf    = {50,50},
10596   /four.osf   = {50,50},
10597   /seven.osf  = {50,50},
10598     IV   = {  , 35},
10599     VI   = { 35,  },
10600     VII  = { 30,  },
10601     VIII = { 25,  },
10602     IX   = {  , 35},
10603     XI   = { 35,  },
10604     XII  = { 30,  },
10605     iv   = {  , 25},
10606     vi   = { 25,  },
10607     vii  = { 20,  },
10608     viii = { 20,  },
10609     ix   = {  , 25},
10610     xi   = { 25,  },
10611     xii  = { 20,  },
10612   % textcomp
10613   \textquotesingle = {400,500},
10614     _ = {200,250},
```

```
10615      ƒ = {    ,100},
10616      ₵ = { 50,    },
10617      † = {100,100},
10618      ‡ = { 80, 80},
10619      • = {    ,100},
10620      · = {300,400}, % periodcentered
10621      /twodotenleader = {150,200},
10622      /ellipsis  = {100,150},
10623      ℃ = { 80,    },
10624      ° = {400,400},
10625      ™ = {100,100},
10626      © = {100,100},
10627      ® = {100,100},
10628      ª = {200,200},
10629      º = {200,200},
10630      ¹ = {200,200},
10631      ² = {200,200},
10632      ³ = {200,200},
10633      ¬ = {200,    },
10634      ¶ = {    ,100},
10635      − = {300,300}, % minus
10636      ± = {150,200},
10637      × = {100,150},
10638      ÷ = {150,200},
10639      € = { 50,100},
10640      ¥ = { 50, 50},
10641   % Greek
10642      Γ = {    ,150},
10643      Δ = {100,100},
10644      Θ = { 50, 50},
10645      Λ = {100,100},
10646      Ξ = { 50, 50},
10647      Υ = {100,100},
10648      Φ = { 50, 50},
10649      Ψ = { 50, 50},
10650      Ω = {    , 50},
10651      ζ = {    , 50},
10652      λ = { 50, 50},
10653      γ = { 50, 50},
10654      π = { 50, 50},
10655      ρ = {    , 50},
10656      σ = { 50, 50},
10657      τ = { 50, 50},
10658      χ = { 50, 50},
10659      φ = { 50, 50},
10660      ϱ = { 50, 50},
10661      ψ = { 50, 50},
10662   % Cyrillic
10663      Г = {    ,150},
10664      Д = { 50, 50},
10665      Ж = { 50, 50},
10666      К = {    , 50},
10667      Л = { 50,    },
10668      Љ = { 50, 50},
10669      З = { 50, 50},
10670      У = { 50,100},
10671      Ф = { 50, 50},
10672      Ч = { 70,    },
10673      Я = { 50,    },
10674      Ъ = { 50, 50},
10675      Ь = {    , 50},
10676      ж = { 50, 50},
10677      ф = { 50, 50},
10678      ъ = { 50, 50},
10679      Ѱ = { 50, 50},
```

```
10680      r = {    , 50},
10681      V = { 50, 50},
10682    % other
10683      Þ = {    , 50},
10684      þ = {    , 50},
10685      Λ = {100,100},
10686      (ɪ)  = { 35, 65},
10687      (a)  = { 30, 60},
10688    }
10689
10690  \SetProtrusion
10691    [ name      = EBGaramond-it-TU,
10692      load      = EBGaramond-it-T1-LF ]
10693    { encoding = {TU,EU1,EU2},
10694      family   = EBGaramond,
10695      shape    = it }
10696    {
10697      /zero.tosf  = {150,150},
10698      /one.tosf   = {150,150},
10699      /two.tosf   = {80,80},
10700      /three.tosf = {50,80},
10701      /four.tosf  = {50,80},
10702      /five.tosf  = {50,80},
10703      /six.tosf   = {50,50},
10704      /seven.tosf = {50,100},
10705      /eight.tosf = {50,50},
10706      /nine.tosf  = {50,80},
10707      /one.lf     = {50,50},
10708      /two.lf     = {50,50},
10709      /three.lf   = {80,50},
10710      /four.lf    = {50,50},
10711      /five.lf    = {50,50},
10712      /six.lf     = {50,50},
10713      /seven.lf   = {50,50},
10714      /eight.lf   = {50,50},
10715      /nine.lf    = {50,   },
10716      /one.osf    = {50,50},
10717      /two.osf    = {50,50},
10718      /three.osf  = {   ,80},
10719      /four.osf   = {50,50},
10720      /seven.osf  = {50,50},
10721    % textcomp
10722      \textquotesingle = {800,100},
10723      − = {300,300}, % minus
10724      ‗ = {200,250},
10725      † = {200,100},
10726      ‡ = { 80, 80},
10727      • = {300,   },
10728      ℃ = {200,   },
10729      ƒ = {100,   },
10730      ¢ = {100,   },
10731      ™ = {200,   },
10732      © = {200,100},
10733      ® = {200,100},
10734      ¬ = {300,   },
10735      ° = {500,100},
10736      ± = {200,150},
10737      ¹ = {300,100},
10738      ² = {300,   },
10739      ³ = {300,   },
10740      · = {300,500}, % periodcentered
10741      /twodotenleader = {150,300},
10742      /ellipsis  = {100,200},
10743      € = {100,   },
10744      × = {200,100},
```

```
10745      ÷ = {200,200},
10746      ¶ = {    ,100},
10747      ª = {200,200},
10748      º = {200,200},
10749      ¥ = { 50, 50},
10750   % Greek
10751      Δ = {150,    },
10752      Θ = { 50,    },
10753      Λ = {150,    },
10754      Υ = {100, 50},
10755      Φ = { 50,    },
10756      X = { 50,    },
10757      Ψ = {100,    },
10758      Ω = { 50,    },
10759      γ = {    , 50},
10760      λ = { 50,    },
10761   % Cyrillic
10762      У = { 50,    },
10763      Ч = {100,    },
10764      З = {100,    },
10765   % other
10766      Þ = { 50, 50},
10767      þ = {    , 50},
10768   }
10769
10770 \SetProtrusion
10771   [ name    = EBGaramond-sc-TU,
10772     load    = EBGaramond-TU ]
10773   { encoding = {TU,EU1,EU2},
10774     family  = EBGaramond,
10775     shape   = sc }
10776   {
10777      a = {50,50},
10778     \ae = {50,  },
10779      d = {  ,50},
10780      f = {  ,50},
10781      g = {50,  },
10782      j = {50,  },
10783      l = {  ,50},
10784      o = {50,50},
10785     \oe = {50,  },
10786      q = {50,70},
10787      r = {  , 0},
10788      t = {50,50},
10789      y = {50,50},
10790   % Greek
10791      α = {50,50},
10792      γ = {  ,50},
10793      δ = {50,50},
10794      λ = {50,50},
10795      ο = {50,50},
10796      τ = {50,50},
10797      υ = {50,50},
10798      ψ = {50,50},
10799   % Cyrillic
10800      т = {50,50},
10801   }
10802
10803 \SetProtrusion
10804   [ name    = EBGaramond-scit-TU,
10805     load    = EBGaramond-it-TU ]
10806   { encoding = {TU,EU1,EU2},
10807     family  = EBGaramond,
10808     shape   = scit }
```

```
10809    {
10810       a = {50,50},
10811     \ae = {50,  },
10812       d = {  ,50},
10813       f = {  ,50},
10814       g = {50,  },
10815       j = {50,  },
10816       l = {  ,50},
10817       o = {50,50},
10818     \oe = {50,  },
10819       q = {50,70},
10820       r = {  , 0},
10821       t = {50,50},
10822       y = {50,50},
10823    % Greek
10824       α = {50,50},
10825       γ = {  ,50},
10826       δ = {50,50},
10827       λ = {50,50},
10828       ο = {50,50},
10829       τ = {50,50},
10830       υ = {50,50},
10831       ψ = {50,50},
10832    % Cyrillic
10833       т = {50,50},
10834    }
```
10835 ⟨/EBGaramond⟩

3.2.4 Palatino

10836 ⟨*Palatino⟩
```
10837 \SetProtrusion
10838    [ name     = palatino-default ]
10839    { encoding = {TU,EU1,EU2},
10840      family   = {Palatino} }
10841    {
10842       A = {50,50},
10843       D = {  ,50},
10844       J = {50,  },
10845       K = {  ,50},
10846       L = {  ,50},
10847       O = {25,  },
10848       T = {50,50},
10849       V = {50,50},
10850       W = {50,50},
10851       X = {50,50},
10852       Y = {50,50},
10853       b = {  ,25},
10854       d = {25,30},
10855       f = {  ,50},
10856       g = {  ,100},
10857       k = {  ,50},
10858       p = {  ,50},
10859       q = {50,  },
10860       r = {  ,50},
10861       t = {  ,50},  � = {  ,50},  � = {  ,50},
10862       v = {75,50},
10863       w = {50,50},
10864       x = {50,50},
10865       y = {50,70},
10866       1 = {100,50},
10867       2 = {25,50},
10868       4 = {50,  },
10869       6 = {50,  },
10870       9 = {25,  },
```

```
10871        Æ  =  {100,  },
10872        Œ  =  {25,   },
10873        .  =  {  ,700},            ..  =  {  ,350},      ...  =  {,150},
10874       {,}=  {  ,500},
10875        :  =  {  ,500},
10876        ;  =  {  ,500},
10877        !  =  {  ,100},            !!  =  {  ,100},
10878        ?  =  {  ,200},            ¿  =  {  ,200},
10879        @  =  {50,50},
10880        ~  =  {200,250},
10881        &  =  {50,100},
10882       \%  =  {100,100},
10883        *  =  {200,200},
10884        +  =  {250,250},
10885        (  =  {100,   },       )  =  {    ,300},
10886        /  =  {200,300},
10887        -  =  {400,500},
10888       \textendash       =  {300,300},     \textemdash        =  {200,200},
10889       \textquoteleft    =  {500,700},     \textquoteright    =  {500,700},
10890       \textquotedblleft =  {300,400},     \textquotedblright =  {300,400},
10891       \textbackslash    =  {200,300},
10892       \quotesinglbase   =  {400,400},     \quotedblbase      =  {400,400},
10893       \guilsinglleft    =  {400,400},     \guilsinglright    =  {300,500},
10894       \guillemotleft    =  {300,300},     \guillemotright    =  {200,400},
10895       \textexclamdown   =  {100,   },     \textquestiondown  =  {100,    },
10896       \textbraceleft    =  {400,200},     \textbraceright    =  {200,400},
10897       \textless         =  {200,100},     \textgreater       =  {100,200},
10898       ≤               =  {200,100},     ≥                  =  {100,200},
10899       \textminus              =  {300,300},
10900       \texttrademark          =  {200,200},
10901       \textcopyright          =  {200,200},
10902       \textregistered         =  {200,200},
10903       \textdegree             =  {300,300},
10904       ¦               =  {450,500},     ¬                  =  {250,150},
10905       ⌐               =  {150,250},
10906       ·                       =  {850, 700},
10907       ¶                       =  {100,0},
10908       ×                       =  {150, 300},
10909       ª               =  {300,300},     º                  =  {300,300},
10910       ⁰  =  {200,400},
10911       ¹  =  {400,350},            ²  =  {200,300},           ³  =  {250,400},
10912       ⁴  =  {250,350},            ⁵  =  {200,300},           ⁶  =  {250,400},
10913       ⁷  =  {200,450},            ⁸  =  {250,400},           ⁹  =  {200,350},
10914       ₀  =  {200,400},
10915       ₁  =  {400,250},            ₂  =  {200,300},           ₃  =  {250,400},
10916       ₄  =  {250,350},            ₅  =  {200,300},           ₆  =  {250,400},
10917       ₇  =  {200,450},            ₈  =  {250,400},           ₉  =  {200,350},
10918       ±  =  {150,100},                                       ÷  =  {300,300},
10919       þ  =  {  ,25},
10920       ̦   =  {300,450},            ̠   =  {300,450},
10921       ̓   =  {300,450},            ̄   =  {300,450},
10922       †               =  {200,250},     ‡                  =  {200,250},
10923       π  =  {50,   },
10924       ƒ  =  {  ,50},
10925       №  =  {100,150},
10926       \textservicemark        =  {100,200},
10927       -  =  {400,500},            -  =  {400,500},           –  =  {200,300},
10928       –  =  {205,305},            —  =  {200,300},           —  =  {50,150},
10929       •  =  {125,200},
10930 %      /a.sc  =  {50,50},

10931    }

10932
10933 \SetProtrusion
10934    [ name      = palatino-it   ]
10935    { encoding = {TU,EU1,EU2},
```

```
10936      family   = {Palatino},
10937      shape    = {it,sl}  }
10938    {
10939      A = {50,50},
10940      Æ = {50,  },
10941      B = {50,  },
10942      C = {50,  },
10943      D = {50,50},
10944      E = {50,  },
10945      F = {50,  },
10946      G = {50,  },
10947      H = {50,  },
10948      K = {50,  },
10949      L = {50,  },
10950      O = {50,  },
10951      Œ = {50,  },
10952      P = {50,  },
10953      Q = {50,  },
10954      R = {50,  },
10955      S = {50,  },
10956      $ = {50,  },
10957      T = {100, },
10958      U = {50,  },
10959      V = {100,50},
10960      W = {50,  },
10961      X = {50,  },
10962      Y = {100,50},
10963      b = {  ,50},
10964      c = {25,  },
10965      g = {75,  },
10966      i = {25,  },
10967      m = {  ,50},
10968      n = {  ,50},
10969      p = {  ,25},
10970      q = {25,  },
10971      x = {  ,50},
10972      1 = {100, },
10973      2 = {50,  },
10974      4 = {50,  },
10975      7 = {50,  },
10976      . = {  ,500},         .. = {  ,350},      ... = {  ,200},
10977    {,}= {  ,500},
10978      : = {  ,300},
10979      ; = {  ,300},
10980      ? = {  ,300},         ¿ = {  ,300},
10981      & = {50,50},
10982      \% = {100,100},
10983      * = {200,200},
10984      + = {150,200},
10985      @ = {50,50},
10986      ~ = {200,150},
10987      ( = {200, },       ) = {  ,200},
10988      / = {100,200},
10989      - = {300,500},
10990      \textendash       = {300,300},   \textemdash       = {200,200},
10991      \textquoteleft    = {700,400},   \textquoteright    = {700,400},
10992      \textquotedblleft = {500,300},   \textquotedblright = {500,300},
10993      _ = {100,100},
10994      \textbackslash    = {100,200},
10995      \quotesinglbase   = {500,500},   \quotedblbase      = {400,400},
10996      \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
10997      \guillemotleft    = {300,300},   \guillemotright    = {300,300},
10998      \textexclamdown   = {100, },     \textquestiondown  = {200,   },
10999      \textbraceleft    = {200,100},   \textbraceright    = {200,200},
11000      \textless         = {300,100},   \textgreater       = {200,100},
```

```
11001      ≤                        = {200,100},    ≥                        = {100,200},
11002      ¦                        = {450,500},    ¬                        = {250,150},
11003      ·                           = {850, 700},
11004      ¶                             = {100,0},
11005      ×                             = {150, 300},
11006      ª = {300,250},           ° = {300,300},           º = {300,250},
11007      ⁰ = {300,200},
11008      ¹ = {300,150},           ² = {350,200},           ³ = {250,150},
11009      ⁴ = {350,100},           ⁵ = {300, 50},           ⁶ = {400,100},
11010      ⁷ = {400, 50},           ⁸ = {250, 50},           ⁹ = {300, 50},
11011      ₀ = {300,300},
11012      ₁ = {300,350},           ₂ = {300,150},           ₃ = {250,250},
11013      ₄ = {400,200},           ₅ = {300,100},           ₆ = {450,200},
11014      ₇ = {450,150},           ₈ = {400,250},           ₉ = {400,200},
11015      ± = {150,100},                                    ÷ = {300,300},
11016      þ = {  50,    },
11017      †                        = {250,200},    ‡                        = {250,200},
11018      ₊ = {300,450},           ₋ = {300,450},
11019      ˖ = {300,450},           ˗ = {300,450},
11020      - = {300,500},           ‐ = {300,500},           – = {100,300},
11021      – = {125,305},           — = {200,300},           — = {125,150},
11022      • = {125,200}
11023    }
11024
11025 \SetProtrusion
11026    [ name    = palatino-sc,
11027      load    = palatino-default ]
11028    { encoding = {TU,EU1,EU2},
11029      family   = {Palatino},
11030      shape    = sc }
11031    {
11032      a = {50,50},
11033      æ = {50,   },
11034      b = { 0, 0},
11035      d = { 0, 0},
11036      f = { 0, 0},
11037      g = { 0, 0},
11038      j = {50,   },
11039      l = {   ,50},
11040      o = { 0, 0},
11041      p = { 0, 0},
11042      q = { 0,   },
11043      r = {   , 0},
11044      t = {50,50},
11045      y = {50,50},
11046      fl = { 0,50},
11047      ffl = { 0,50},
11048      ◆ = { 0,50},
11049      ◆ = { 0,50}
11050    }
11051 ⟨/Palatino⟩
```

3.2.5   Basic glyph set

The protrusion settings will still be loaded from microtype.cfg.

```
11052 ⟨TU-basic⟩ %% No settings.
```

3.2.6   Empty glyph set

```
11053 ⟨*TU-empty⟩
11054 \SetProtrusion
11055    [ name = empty ]
11056    { encoding = {TU,EU1,EU2},
```

```
11057      family   = {TU-empty} }
11058    { }
11059 ⟨/TU-empty⟩
11060
```

4 Auxiliary file for micro fine tuning

This file may be used to test protrusion and (less so) expansion settings.

```
11061 ⟨*test⟩
11062 \documentclass{article}
11063 %% options are passed through to microtype
11064 \usepackage[stretch=50]{microtype-show}
11065
11066 %% options for microtype-show
11067 \ShowGlyphIndextrue
11068 \ShowMissingGlyphstrue
11069 \def\GlyphScaleFactor{2}
11070
11071 %% load any required font packages:
11072 \ifpdftex
11073  \usepackage[T1]{fontenc}
11074 \else
11075  \usepackage{fontspec}
11076 \fi
11077
11078 \begin{document}
11079 \microtypesetup{expansion=false}
11080
11081 %% load your font here:
11082
11083 \ShowCharacterInheritance
11084
11085 \newpage
11086 \ShowProtrusion
11087
11088 \newpage
11089 %% show single glyphs
11090  %\ShowDummyLine
11091  %\ShowProtrusionLineGlyph{A}
11092  %\ShowProtrusionLineIndex{27}
11093
11094 %% loop through all glyphs of the font;
11095 %% protrusion values are shown in 1000th of 1em
11096  \ShowProtrusionDefined
11097
11098  %\ShowProtrusionMissing
11099
11100  %\ShowProtrusionAll
11101
11102 \newpage %% ------------------------------------------------------------------
11103 This is the current font stretched by 5\%, normal, and shrunk by 5\%:
11104
11105 \newlength{\MTln}
11106 \newcommand*\teststring
11107   {ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}
11108 \settowidth{\MTln}{\teststring}
11109 \microtypesetup{expansion=true}
11110
11111 \bigskip\noindent\parbox{1.05\MTln}{\teststring\linebreak\\\teststring}\par
11112 \bigskip\noindent\parbox{0.95\MTln}{\teststring}
11113 \end{document}
11114 ⟨/test⟩
```

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net or file an issue at https://github.com/schlcht/microtype/issues.

A The title logo

This is `microtype-logo.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a `dtx` file
- `\input` it in the preamble: it then provides the command `\printlogo`, which will do just that

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

11115 ⟨*logo⟩

Here's how the logo on the title page was created. [19] It has nothing to do with `microtype`, actually, but uses `fontinst`. It is based on an experiment I posted to the de.comp.text.tex newsgroup. [20] It will show:

- the character
- the TEX box
- the bounding box
- kerns

A.1 Macros

To run this file, TEX needs to find the `afm` file (either in the `TEXINPUTS` path, or in the current working directory). First input `fontinst`.

11116 `\input fontinst.sty`

`bbox.sty` is an addition to `fontinst`, which makes dimensions of the bounding boxes available (and was written by Hàn Thế Thành, by the way). These dimensions are specified in the `afm` file, but not used by TEX, which is why `fontinst` will discard them otherwise.

11117 `\input bbox.sty`

`\tempdim`   Allocate some dimen registers.

11118 `\newdimen\tempdim`

`\fboxrulei`   Frame width of the box as TEX sees it.

11119 `\newdimen\fboxrulei`
11120 `\fboxrulei=0.1pt`

`\fboxruleii`   Frame width of the bounding box.

11121 `\newdimen\fboxruleii`
11122 `\fboxruleii=0.1pt`

`\kernboxheight` Height of the box indicating the kern.

11123 `\newdimen\kernboxheight`
11124 `\kernboxheight=5pt`

`\scaletoem`   An auxiliary macro. Return a dimension relative to the em-width of the font. Requires e-TEX.

11125 `\setcommand\scaletoem#1{\dimexpr #1 sp*\fontdimen6\font/1000\relax}`

`\showlogo`   A `fontinst` incantation whose sole purpose is to produce the logo. Its argument is a string (letters only).

11126 `\fontinstcc`
11127 `\def\showlogo#1{%`

Some fonts do not specify the `\fontdimen 6` (width of an em) in the `afm` file. In this case, use the font size, which is correct in most cases.

11128 `  \ifdim\fontdimen6\font = 0pt`
11129 `    \typeout{***~Warning:~no~fontdimen~6~specified~***^^J%`
11130 `            ***~setting~it~to~\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi~***}`
11131 `    \fontdimen6\font=\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi\relax`
11132 `  \fi`
11133 `  \installfonts`
11134 `    \input_metrics{}{\logofont,\metrics\printbbs{#1}\relax}`

---

19   Note that the `logo` module will not be created when installing `microtype`. Instead, the source file `microtype-logo.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the `pdftk` tool.

20   Message ID: `42aa3687$0$24366$9b4e6d93@newsread2.arcor-online.net`

```
11135     \endinstallfonts
11136 }
11137 \normalcc
```

Layers.

```
11138 \makeatletter
11139 \def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
11140 \ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
11141 \ifx\mt@order  \@undefined\let\mt@order  \@empty\fi
11142 \xdef\mt@order{\mt@order[(Logo)}
11143 \let\mtl@resources\@empty
11144 \def\mtl@register#1{%
11145     \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
11146     \expandafter\xdef\csname mtl@#1\endcsname{\the\pdflastobj\space 0 R }
11147     \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
11148     \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
11149     \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
11150 \mtl@register{canvas}
11151 \mtl@register{characters}
11152 \mtl@register{bounding-boxes}
11153 \mtl@register{TeX-boxes}
11154 \xdef\mt@order{\mt@order]}
11155 \global\let\mtl@objects\mt@objects
11156 \def\togglelayer#1#2{%
11157     \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
11158       user{/Subtype/Link
11159             /BS << /Type/Border/W 0 >> /H/O
11160             /A << /S/SetOCGState
11161                   /State[/Toggle \csname mtl@#1\endcsname] >>
11162     }#2\pdfendlink
11163 }
```

`\printbbs`   Preparation.

```
11164 \setcommand\printbbs#1{%
11165     \setbox0\hbox{#1}%
11166     \leavevmode
11167     \kern-\fboxrulei
```

The canvas in the natural width of the text minus protrusion, in color bgcolor.

```
11168     \mtl@layer{canvas}{%
11169       \getboundarychars#1\relax
11170       \tempdim=\dimexpr\wd0 - (\scaletoem{\lpcode\font\firstchar}+
11171                             \scaletoem{\rpcode\font\lastchar})\relax
11172       \kern\dimexpr\scaletoem{\lpcode\font\firstchar}\relax
11173       \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
11174             \hrule width  \tempdim
11175                   height \dimexpr\dp0+\ht0+0.15em\relax}%
11176       \kern-\tempdim
```

The baseline, in color blcolor.

```
11177       \vbox{\color{blcolor}%
11178             \hrule width  \tempdim
11179                   height \fboxrulei}%
11180     }%
11181     \kern-\dimexpr\wd0 -\scaletoem{\rpcode\font\lastchar}\relax
```

The string.

```
11182     \printbbss #1\relax\relax
11183 }
```

`\getboundarychars`   Get first . . . .

```
11184 \def\getboundarychars#1#2\relax{%
11185     \def\firstchar{`#1}%
11186     \getlastchar#1#2\relax
11187 }
```

`\getlastchar`   . . . and last character.

```
11188 \def\getlastchar#1#2{%
```

```
11189        \ifx\relax#2\relax
11190          \def\lastchar{`#1}%
11191        \else
11192          \expandafter\getlastchar
11193        \fi #2%
11194  }
```

\printbbss   Loop over all characters of the string.

```
11195  \def\printbbss#1#2#3\relax{%
11196    \ifx\relax#1\relax
11197    \else
11198      \ifx\relax#2\relax
11199        \printbb{#1}{}%
11200      \else
11201        \printbb{#1}{#2}%
11202      \fi
11203      \expandafter\printbbss
11204    \fi #2#3\relax
11205  }
```

\printbb   Record the kern between the current and the following character, then print the character. `\kerning` is a `fontinst` command.

```
11206  \setcommand\printbb#1#2{%
11207    \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
11208    \showboxes{#1}%
```

This could be another application.

```
11209  %      \quad
11210  %      w: \the\scaletoem{\width{#1}},
11211  %      bb: \the\scaletoem{\bbleft{#1}}/%
11212  %          \the\scaletoem{\bbright{#1}},
11213  %          \the\scaletoem{\number\numexpr\width{#1}-\bbright{#1}\relax}
11214  %      h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
11215  }
```

\showboxes   Print the boxes for char ⟨#1⟩. This won't work if ⟨#1⟩ isn't also the PostScript name of the glyph (e.g., 'comma' ≠ ',').

```
11216  \setcommand\showboxes#1{%
11217    \leavevmode
11218    \color{texcolor}%
```

We have to record the width of the glyph.

```
11219    \setbox0\hbox{{\color{textcolor}#1}}%
11220    \global\tempdim=\wd0\relax
11221    \kern-\fboxrulei
```

1. *The TEX box:* Print a frame in color `texcolor`. This frame shows the glyph as TEX sees it.

```
11222        \mtl@layer{TeX-boxes}{%
11223          \hbox{%
11224            \lower\dimexpr \dp0 + \fboxrulei\relax
11225            \hbox{%
11226              \vbox{%
11227                \hrule height\fboxrulei
11228                \hbox{%
11229                  \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
11230                  \phantom{\unhcopy0}%
11231                  \vrule width\fboxrulei
11232                }%
11233                \hrule height\fboxrulei}}}%
11234        }%
```

2. *The character:* Now we step back and print the actual glyph. We hold it back until now, so that it will be printed *on top* of its box.

```
11235        \kern-\wd0
11236        \mtl@layer{characters}{\hbox{\box0}}%
```

Step back by the amount that the character's bounding box differs from the TEX box on the left side.

```
11237        \kern\dimexpr\scaletoem{\bbleft{#1}}-\tempdim-\fboxruleii\relax
```

3. *The bounding box:* will be printed in color `bbcolor`.

```
11238        \mtl@layer{bounding-boxes}{%
11239          {\color{bbcolor}%
11240          \hbox{%
11241            \lower\dimexpr-\scaletoem{\bbbottom{#1}}+\fboxruleii\relax
11242            \hbox{%
11243              \vbox{%
11244                \hrule height\fboxruleii
11245                \hbox to \dimexpr\scaletoem{\numexpr
11246                            \bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{%
11247                  \vrule height \dimexpr\scaletoem{\numexpr
11248                            \bbtop{#1}-\bbbottom{#1}\relax}%
11249                        width\fboxruleii
11250                  \hfill
11251                  \vrule width\fboxruleii}%
11252                \hrule height\fboxruleii}}}%
11253          }%
11254          \kern-\dimexpr\fboxruleii+\fboxrulei\relax
11255        }%
```

4. *The kern:* We also print a small box in color `kerncolor` indicating the kerning between the current and the next character; filled for negative kerns, empty for positive kerns.

```
11256        \kern\scaletoem{\numexpr\width{#1}-\bbright{#1}\relax}%
11257        \mtl@layer{TeX-boxes}{%
11258          {\ifnum\thekern<0
11259            \color{kerncolor}%
11260            \kern\scaletoem{\thekern}%
11261            \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoem{\thekern}\relax
11262                                      height \kernboxheight}%
11263            \kern\scaletoem{\thekern}%
11264          \else
11265            \color{texcolor}%
11266            \ifnum\thekern=0 \else
11267              \lower\kernboxheight
11268              \hbox{%
11269                \vbox{%
11270      %          \hrule height\fboxrulei
11271                  \hbox{%
11272                    \vrule height \kernboxheight width\fboxrulei
11273                    \kern\dimexpr\scaletoem{\thekern}-2\fboxrulei\relax
11274                    \vrule width\fboxrulei
11275                  }%
11276                \hrule height\fboxrulei}}%
11277            \fi
11278          \fi
11279          }%
11280        }%
11281    %  \kern-\fboxrulei
11282    }
```

`\printlogo`

```
11283 \newbox\logobox
11284 \def\printlogo{%
11285  \setbox\logobox=\hbox{\vbox{%
11286    \MakePercentComment
```

This is the Kepler MM font used in the logo.

```
11287    \def\logofont{pkpri9e10}
11288    \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
11289    \font\thelogofont=\logofont\space at 82pt
```

This would load the italic Palatino font instead.

```
11290 %\def\logofont{pplri}
11291 %\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
11292 %\edef\logofont{\logofont8r}
11293 %\font\thelogofont=\logofont\space at 78pt
```

Load the font.

```
11294      \thelogofont
```

Protrusion values (overdone for didactic reasons).

```
11295      \lpcode\font`M=96
11296      \rpcode\font`e=46
```

Now we can generate the logo.

```
11297      \pdfliteral direct{/SXS gs}%
11298      \showlogo{Microtype}%
11299 %    \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
11300 %    \kern5pt\\[3\baselineskip]
11301 % \long\def\@makefntext##1{%
11302 %    \leftskip 0pt
11303 %    \parindent 0pt
11304 %    \everypar{\parindent 0pt}%
11305 %    \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
11306 % \footnotetext[1]{This graphic displays on a
11307 %    \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
11308 %    their \togglelayer{bounding-boxes}{bounding boxes}
11309 %    and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
11310  }}%
11311  \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
11312  \immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}%
11313  \immediate\pdfxform
11314          attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
11315          resources {/Properties <<\mtl@resources>>
11316                     /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
11317          \logobox
11318 % \vskip-2.5\baselineskip
11319 % \leavevmode
11320 % \togglelayer{characters}{%
11321 %    \pdfrefxform\pdflastxform
11322 % }%
11323  \pdfannot\logodimens{%
11324      /Subtype/Widget /FT/Btn /T(Logo)
11325      %/F 4 % why did I say this?
11326      /AP << /N \the\pdflastxform\space 0 R >>
11327      /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
11328             /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
11329             /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
11330             /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
11331         >> }%
11332  \vspace{3\baselineskip}
11333 }
11334 \IfFileExists{pkpmmri8a10.afm}\relax{\def\printlogo{\MT@warning{File pkpmmri8a10.afm not found.
11335  \MessageBreak Cannot create logo}}}
```

Our font.

```
11336 \pdfmapline{+pkpmmri8r10 KeplMM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
```

Define colours (thered and thegreen are copied from microtype.dtx).

```
11337 \def\mtdefinecolors{
11338  \definecolor{thered}{rgb}{0.65,0.04,0.07}
11339  \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
11340  \colorlet{texcolor}{thegreen!50}   % TeX boxes
11341  \colorlet{kerncolor}{texcolor}     % negative kerns
11342  \colorlet{bbcolor}{thered!50}      % bounding box
11343  \colorlet{bgcolor}{black!8}        % canvas
11344  \colorlet{blcolor}{black!50}       % baseline
11345  \colorlet{textcolor}{black!40}     % text
11346 }
```

Use with microtype.dtx

```
11347 \ifx\documentclass\@twoclasseserror
11348  \usepackage[xcdraw]{xcolor}
11349  \mtdefinecolors
```

11350 `\else`

A.2 Document

Now we can start the document.

11351 `\documentclass[10pt,a4paper]{ltxdoc}`
11352 `\providecommand\MakePercentComment{\relax}`
11353 `\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}`

Re-use the preamble from `microtype.dtx`.

11354 `\usepackage{microtype-doc}`
11355 `\usepackage{attachfile}`
11356 `\makeatletter`
11357 `\pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}`
11358 `\makeatother`
11359 `\begin{document}`

You are currently reading this.

11360 `\DocInput{microtype-logo.dtx}`
11361 `\newpage`
11362 `And here it is:\vspace{6\baselineskip}`
11363 `\begin{center}`
11364 `  \printlogo`
11365 `\end{center}`
11366 `\expandafter\enddocument`
11367 `\fi`

That's it.

11368 ⟨*/logo*⟩

B The letterspacing illustration

This is `microtype-lssample.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a `dtx` file
- `\input` it in the preamble: it then provides the commands
  - `\lssample`: prints the letterspacing illustration
  - `\anchorarrow`: anchors an arrow for layer ⟨*#1*⟩
  - `\showarrow`: toggles layer ⟨*#1*⟩ or ⟨*#2*⟩, and prints ⟨*#2*⟩

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

11369 `\ifx\lssample\undefined`
11370 ⟨**lssample*⟩

Upon popular request, here's how I've created the letterspacing illustration. [21]

B.1 Macros

Rule width and image height and depth.

11371 `\makeatletter`
11372 `\newdimen\lsamount`
11373 `\newdimen\lsrule`
11374 `\lsrule=0.2pt`
11375 `\def\lsheight{8pt}`
11376 `\def\lsdepth{12pt}`

---

21 Note that the `lssample` module will not be created when installing `microtype`. Instead, the source file `microtype-lssample.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the `pdftk` tool.

Our font (Adobe Caslon).

```
11377 \def\lsfont{\fontfamily{paca}\selectfont}
```

Loop over all letters in ⟨#2⟩, letterspacing them by ⟨#1⟩.

```
11378 \def\dols#1#2{\lsamount=#1\relax \dolss#2\enddols}
11379 \def\dolss#1#2\enddols{%
11380     \ifx\empty#2\empty\divide\lsamount 2\fi
11381     \ls{#1}%
11382     \ifx\empty#2\empty\else \dolss#2\enddols \fi
11383 }
```

One `tikz` picture for each letter.

```
11384 \def\ls#1{%
11385     \begin{tikzpicture}[remember picture,line width=\lsrule]
11386     \tikzstyle{every node}=[inner sep=0pt]
```

The bounding box.

```
11387     \mts@layer{stuff}{%
11388       \node[draw=thegrey,
11389             fill=theshade,
11390             outer sep=\lsrule,
11391             anchor=base,
11392             font=\lsfont]{\phantom{#1}};
11393     }
```

The letter.

```
11394     \node[anchor=base,font=\lsfont](#1){#1};
```

Two auxiliary coordinates.

```
11395     \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
11396     \path (#1.base east)  ++(-.5\lsrule,-\lsdepth)  coordinate (#1R);
11397     \mts@layer{stuff}{%
```

Now draw the normal character width,

```
11398     \draw[color=thered!75,
11399           fill=thered!30,
11400           outer sep=\lsrule]
11401           (#1L) rectangle (#1R);
11402     \ifdim\lsamount>0pt
11403       \path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
11404       \path (#1R) ++(\lsamount+\lsrule,+\lsdepth) coordinate (#1E);
```

and the letter space.

```
11405       \draw[color=thered,
11406             fill=thered!50,
11407             outer sep=\lsrule]
11408             (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
11409     \fi
11410     }
11411   \end{tikzpicture}%
11412   \ignorespaces
11413 }
```

Draw the interword space.

```
11414 \def\lssp#1#2#3#4{%
11415   \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
11416     \mts@layer{stuff}{%
11417       \tikzstyle{every draw}=[anchor=bottom]
11418       \coordinate(#1space) at (#2/2,\lsdepth/2);
11419       \coordinate(#1stretch) at (#2+#3/2,+0pt);
11420       \coordinate(#1shrink) at (#2-#4/2,+0pt);
11421       \draw[color=thegreen,fill=thegreen!50,use as bounding box]
11422             (0,0) rectangle ++(+#2,+\lsdepth);
11423       \draw[color=thegreen,fill=thegreen!30]
11424             (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
11425       \draw[color=thegreen,fill=thegreen!50]
11426             (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
11427       \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
```

```
11428            (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
11429         \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
11430            (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
11431     }%
11432   \end{tikzpicture}%
11433   \ignorespaces
11434 }
```

Layers.

```
11435 \def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
11436 \def\mtsx@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
11437 \ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
11438 \ifx\mt@order  \@undefined\let\mt@order  \@empty\fi
11439 \xdef\mt@order{\mt@order[(Sheep)}
11440 \let\mts@resources\@empty
11441 \def\mts@register#1{%
11442   \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
11443   \expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj\space 0 R }
11444   \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
11445   \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
11446   \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
11447 \mts@register{stuff}
11448 \mts@register{tracking}
11449 \mts@register{ispace}
11450 \mts@register{ospace}
11451 \mts@register{istretch}
11452 \mts@register{ishrink}
11453 \mts@register{ostretch}
11454 \mts@register{oshrink}
11455 \mts@register{okern}
11456 \mts@register{ligature}
11457 \mts@register{_compatibility}
11458 \xdef\mt@order{\mt@order]}
```

Anchor point for the arrow in the code.

```
11459 \newcommand\anchorarrow[1]{%
11460   \tikz[remember picture,overlay]\node(#1_c){};}
```

Add an arrow from code to image.

```
11461 \newcommand\add@arrow[5][left]{%
11462   \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
11463     \mtsx@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
11464 }
```

Toggle layer.

```
11465 \def\toggle@layer#1#2#3{%
11466   \pdfstartlink
11467     user{/Subtype/Link
11468         /BS << /Type/Border/W 0 >> /H/O
11469 %       /BS << /Type/Border/W 1 /S/D /D[4 1] >>
11470 %       /C[0.7 0.7 0.7] /H/O
11471         /Contents(Click to Toggle!)
11472         /A << /S/SetOCGState
11473               /State[/Toggle \csname mts@#1\endcsname] >> }%
11474   \rlap{#2}%
11475   {\fboxsep=0pt \fboxrule=0pt
11476    \mtsx@layer{stuff}{%
11477      \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
11478    \mtsx@layer{#1}{%
11479      \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}%
11480   }%
11481   \pdfendlink
11482 }
11483 \newcommand\showarrow[2][]{%
11484   \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
11485   \toggle@layer{\@tempa}{{\itshape #2}}}
```

The environment for our illustration.

```
11486 \def\ls@sample#1{{%
11487   \parskip  4pt \parindent 0pt
11488   \par
11489   \vskip4pt
11490   {\leftskip 15pt
11491   \mt@pseudo@marg{\color{theblue}Click on the image to show the kerns
11492      and spacings involved. Click on emphasised words in the text below
11493      to reveal the relation of image and code.\strut}
11494   \mt@layer{_compatibility}{%
11495      \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
11496        \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
11497      \mt@pseudo@marg{\color{thered}%
11498        If you had a \acronym{PDF} viewer that understands
11499        \acronym{PDF}\,{\smaller1.5}, you could hide the arrows selectively.}}
11500   \vskip-\mt@unvdimen}%
11501   \vskip-4pt
11502   \setlength\fboxsep{4pt}%
11503   \leavevmode
11504   \pdfstartlink
11505      user{/Subtype/Link
11506          /BS << /Type/Border/W 0 >> /H/O
11507          /A << /S/SetOCGState
11508              /State[/Toggle \mts@stuff] >> }%
11509      \fcolorbox{theframe}{theshade}%
11510        {\fontsize{34}{38}\selectfont #1}%
11511   \pdfendlink
11512   \par\medskip
11513   }%
11514   \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
11515 }
```

Now define the illustration to be used in the document.

```
11516 \def\lssample{%
11517   \ls@sample{%
11518     \dols{0pt}{Stop}
11519       \lssp{o}{0.45em}{0.25em}{0.15em}
11520     \dols{0.16em}{{st}ealing}\hskip-\dimexpr 0.08em+\lsrule\relax
11521       \lssp{i}{13.82pt}{4.65pt}{2.08pt}
11522     \dols{0.16em}{sheep}
11523     \dols{0pt}{!}
11524   }%
```

Don't forget to add the arrows.

```
11525   \vspace{-\baselineskip}
11526   \add@arrow{red}       {tracking}{lsamount_c.east}{a_ls}
11527   \add@arrow{red}       {okern}   {okernend_c.east}{p_ls}
11528   \add@arrow{green}     {ospace}  {ospace_c.east}   {ospace}
11529   \add@arrow{green}     {ispace}  {ispace_c.center}{ispace}
11530   \add@arrow{green!75}  {istretch}{istretch_c.east}{istretch.north}
11531   \add@arrow{green!75}  {ishrink} {ishrink_c.west} {ishrink.north}
11532   \add@arrow{green!75}  {ostretch}{ostretch_c.east}{ostretch.north}
11533   \add@arrow{green!75}  {oshrink} {oshrink_c.east} {oshrink.north}
11534   \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
11535 }
11536 \fi
```

This is for use with microtype.dtx

```
11537 \ifx\documentclass\@twoclasseserror
11538   \usepackage{tikz}
11539 \else
```

B.2 Document

```
11540 \documentclass[10pt,a4paper]{ltxdoc}
11541 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
```

Re-use the preamble from `microtype.dtx`.

```
11542 \usepackage{microtype-doc}
11543 \usepackage{attachfile}
11544 \usepackage{tikz}
11545 \makeatletter
11546 \pdfcatalog{/OCProperties << /OCGs [\mt@objects]
11547                             /D << /Order [\mt@order] /BaseState/OFF >> >> }
11548 \makeatother
11549 \begin{document}
```

You are currently reading this.

```
11550 \DocInput{microtype-lssample.dtx}
```

Now show what we are able to do.

```
11551 \noindent
11552  Since a picture is worth a thousand words, probably even more if, in our
11553  case, it depicts a couple of letterspaced words, let's bring one to sum up
11554  these somewhat confusing options. Suppose you had the following settings
11555  (which I would in no way recommend; they are only for illustrative purposes):
11556 \begin{verbatim}
11557 \SetTracking
11558   [ no ligatures  = {"\anchorarrow{nolig}"f},
11559     spacing       = {60"\anchorarrow{ispace}"0*,%
11560                       "-1"\anchorarrow{istretch}"00*, "\anchorarrow{ishrink}"},
11561     outer spacing = {4"\anchorarrow{ospace}"50,"%
11562                       "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
11563     outer kerning = {"\anchorarrow{okernbegin}"*,"%
11564                       \anchorarrow{okernend}"*} ]
11565   { encoding = * }
11566   { 1"\anchorarrow{lsamount}"60 }
11567 \end{verbatim}
11568  and then write:
11569 \begin{verbatim}
11570 Stop \textls{stealing sheep}!
11571 \end{verbatim}
11572  this is the (typographically dubious) outcome:
11573
11574  \lssample
11575
11576 \noindent
11577  While the word `Stop' is not letterspaced, the space between the letters in
11578  the other two words is expanded by the \showarrow[tracking]{tracking~amount}{red}
11579  of 160/1000\,em\,=\allowbreak\,0.16\,em.
11580  The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
11581  increased by 60\%, while its \showarrow[istretch]{stretch}{green} amount is
11582  decreased by 10\% and the \showarrow[ishrink]{shrink}{green} amount is left
11583  untouched.
11584  The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
11585  piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
11586  \showarrow[oshrink]{shrink}{green} by 0.15\,em.
11587  Note that there is no outer space after the text, since the exclamation mark
11588  immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
11589  of half the letterspace amount (0.08\,em) is added.
11590  Furthermore, one \showarrow{ligature}{grey} wasn't broken up, because we
11591  neglected to specify the `|s|' in the |no ligatures| key.
11592
11593 \expandafter\enddocument
11594 \fi
```

```
11595 ⟨/lssample⟩
```

C Change history

1.0 1.1 1.2 1.3 1.4 1.4a 1.4b 1.5 1.6 1.6a 1.7 1.8 1.9 1.9a 1.9b 1.9c 1.9d 1.9e 1.9f 2.0 2.1 2.2 2.3 2.3a 2.3b 2.3c 2.3d 2.3e 2.4 2.5 2.5a 2.6 2.6a 2.7 2.7a 2.7b 2.7c 2.7d 2.8 2.8a 2.8b 2.8c 3.0 3.0a 3.0b

2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
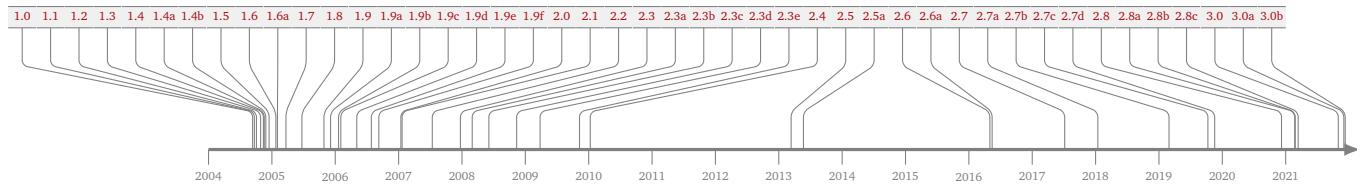
Numbers prefixed with 'U' refer to the User manual.

2004/09/11 Version 1.0

General: Initial version U1

2004/09/21 Version 1.1

General: configuration file names in lowercase (suggested by *Harald Harders*) 71
  remove 8-bit characters from the configuration files (suggested by *Harald Harders*) 133
Protrusion: add factors for some more characters . 141
  settings for Adobe Minion (contributed by *Harald Harders*) 142
\DeclareCharacterInheritance: new command: possibility to specify character inheritance 104
\MT@declare@sets: remove spaces around set name 90
\MT@find@file: fix: also check whether the file for the base font family has already been loaded . . 71
\MT@get@basefamily: only remove suffixes 'x' or 'j' 73
\MT@get@listname@: don't check for empty attributes

list 73
\MT@ifempty: fix: use category code 12 for the percent character (reported by *Tom Kink*) 20
\MT@is@number: numbers may also be specified in hexadecimal or octal (suggested by *Harald Harders*) 79
\MT@pdftex@no: fix: version check (reported by *Harald Harders*) 14
\MT@permute: don't use sets for empty encoding . . 106
\MT@setup@expansion: issue an error instead of a warning, when pdfTEX version is too old for autoexpand 123
\MT@split@codes: fix: allow zero and negative values 42
\MT@use@set: remove spaces around set name 95

2004/10/03 Version 1.2

Font aliases: declare cmor as an alias of cmr 130
Font sets: new: allmath and basicmath 129
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding 172
  add settings for Computer Modern Roman math symbols 176
\MT@familyalias: define alias font name as an alternative, not as a replacement 38
\MT@get@basefamily: also remove 'w' (swash capitals) 73
\MT@get@highlevel: check whether defaults have changed 91

\MT@get@inh@list: fix: set inheritance list \globally to \@empty 75
\MT@get@listname@: alternatively check for alias font name 73
\MT@get@size: additional magic to catch some errors 92
\MT@get@size@@: hijack \set@fontsize instead of \@setfontsize 93
\MT@loop: fix: new macro, used instead of \loop . . 24
\MT@maybe@do: also check for alias font name 39
\MT@permute@@@@@: more sanity checks for \SetProtrusion and \SetExpansion 108
\MT@setupfont: also search for alias font file 35
  fix: call \@@enc@update if necessary 36

2004/10/27 Version 1.3

General: fix: specifying load option does no longer require to give a name, too 101
Font aliases: declare aer, zer and hfor as aliases of cmr 130

\MT@fix@catcode: check some category codes (compatibility with german) 5

\MT@load@list: check whether list exists 71

2004/11/12 Version 1.4

General: check for pdfcprot 30
  don't use scratch registers in global definitions . . 75
  use \pickup@font instead of \define@newfont as

  the hook for \MT@setupfont 84
  use one instead of five counters 25
Protrusion: tweak quote characters for cmr variants

(OT1, T1, lmr) 147
\microtypesetup: fix: set the correct levels, and re-
    member them; warning when enabling an option

disabled in package options 116
\SetExpansion: fix: specifying extra options does no
    longer require to give a name, too 98

2004/11/17 Version 1.4a

General: new option: final 113
\MT@cfg@catcodes: fix: reset some more catcodes

when reading files (reported by *Michael Hoppe*) 72

2004/11/26 Version 1.4b

General: fix: set catcodes before reading global con-
    figuration file (reported by *Christoph Bier*) . . 115
    optimisation: use less \expandafters and \csnames 19
Protrusion: harmonise dashes in upshape and italic
    (cmr, pad, ppl) 141
    slanted like italics 152
\MT@checklist@family: fix: don't try alias family
    name if encoding failed 40
\MT@get@basefamily: fix: failed for font names of the

form abczz (reported by *Georg Verweyen*) 73
\MT@get@slot: don't define \MT@char globally (save
    stack problem) 75
\MT@ifdimen: don't set \MT@count globally (save stack
    problem) 21
\MT@setup@PDF: new message if \pdfoutput is
    changed 120
\MT@use@set: don't use undeclared font sets 95

2004/12/15 Version 1.5

General: defaults: step: 4 (suggested by *Hàn Thế
    Thành*) 113
    new option: selected, by default false (suggested
    by *Hàn Thế Thành*) 111
Documentation: add 'Short history' U32
Inheritance: remove \ss from T1 list, add \DJ . . . 134
Protrusion: settings for Bitstream Charter 142
\DeclareMicrotypeAlias: remove spaces around ar-
    guments 96
\MT@cfg@catcodes: reset catcode of '=' (compatibility
    with Turkish babel) 72
\MT@fix@catcode: reset catcode of '^' (compatibility
    with chemsym) 5

\MT@get@highlevel: don't test defaults if called after
    begin document 91
\MT@scale@factor: warning for factors outside limits 45
\MT@scale@to@em: don't use \lpcode and \rpcode for
    the calculation 43
\MT@set@ex@codes: allow non-selected font expan-
    sion 54
\MT@set@pr@codes: adjust protrusion factors before
    setting the inheriting characters 41
\MT@setup@expansion: defaults: calculate step as
    min(stretch,shrink)/5 122
    defaults: turn off expansion for DVI output 121
    disable automatic expansion for DVI output . . . 122

2005/01/24 Version 1.6

General: defaults: turn off expansion for old pdfTEX
    versions 115
    load a font if none is selected 35
    new option: factor, by default 1000 113
    restructure dtx file 129
    test whether \pickup@font has changed 86
    test whether numeric options receive a number 113
    use e-TEX's \ifcsname and \ifdefined if defined 19
Protrusion: add italic uppercase Greek letters . . . 152
    improve settings for numbers (pointed out by *Peter
    Muthesius*) 143

tune CMR math letters (OML encoding) 177
\MT@get@charwd: use e-TEX's \fontcharwd, if available 44
\MT@get@inh@list: correct message if selected is
    false 75
\MT@set@ex@codes: introduce factor option 54
\MT@set@pr@codes: introduce factor option 41
\MT@setup@expansion: disable automatic expansion
    for old pdfTEX versions 123
\MT@use@set: retain current set if new set is un-
    declared 95
\MT@vinfo: new macro instead of \ifMT@verbose . . . 6

2005/02/02 Version 1.6a

Documentation: add table of fonts with tailored pro-
    trusion settings U21
\MT@get@slot: completely redone, hopefully more
    robust (compatible with frenchpro; problem

reported by *Bernard Gaulle*) 75
\MT@pdftex@no: new macro 14
\MT@reset@ef@codes: only reset \efcodes for older
    pdfTEX versions 54

2005/03/23 **Version 1.7**

2005/06/23 **Version 1.8**

2005/10/28 Version 1.9

General: \DeclareMicrotypeSet: new key: font . . 93
  \SetProtrusion: value 'relative' renamed to
    'character' for key unit 102
  allow context-specific font setup 84
  compatibility with TeX Live hack (reported by *Herbert Voß*) 13
  disable microtype setup inside hyperref's
    \pdfstringdef (reported by *Hàn Thế Thành*) . . 30
  fix: use true as the default value 110
  option unit: rename value relative to character 114
Documentation: add hint about verbatim environment U27
  add remark about Type 1 fonts required for automatic font expansion U7
Font aliases: declare qpl and qtm (qfonts, TeX Gyre)
    as aliases of ppl resp. ptm 131
Font sets: add OT4 encoding to text sets 129
  add T5 encoding to text sets 129
Inheritance: add list for OT4 135
  add list for T5 (requested by *Hàn Thế Thành*) . . 136
Protrusion: fix: remove uppercase Greek letters from
    T1 encoded CMR 145
  settings for OT4 encoding (Computer Modern Roman, Palatino, Times) 141

settings for T5 encoded Computer Modern Roman 141
\DisableLigatures: new command: disable ligatures
    (requires pdfTeX 1.30) 97
\microtypecontext: new command: change setup
    context in the document 88
\MT@checklist@family: fix: add two missing
    \expandafters 40
\MT@detokenize@c: fix the ε-TeX version 20
\MT@exp@two@n: new macros: less \expandafters . . 19
\MT@get@opt: new key 'preset' to set all characters to
    the specified value before loading the lists 46
\MT@is@active: redone: use \set@display@protect 79
\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning 78
\MT@maybe@do: redone 39
\MT@rem@from@clist: new macro: remove an item
    from a comma list 23
\MT@scale@factor: generalised 45
\MT@setup@expansion: disable expansion if both step
    and shrink are zero 123
  warning if user requested zero step 122
\MT@toks: use instead of \toks@ 16
\SetProtrusion: (et al.) new key: font 97

2005/12/05 Version 1.9a

General: '⟨file name⟩/⟨line number⟩' as default list
    name 101
  new option: defersetup, by default true 112
  remove superfluous test whether \pickup@font has
    changed 86
Documentation: add explanation for error message
    in DVI mode U28
  add explanation for error message with non-Type 1
    fonts U29
Font aliases: declare mdbch (mathdesign) as an alias
    of bch 132
Protrusion: fix: remove '_' from OT1 encoding . . . 146
  settings for T5 encoded Charter 141
\microtypesetup: inside the preamble, accepts all
    package options 116
\MT@check@font@cx: optimise context-sensitive setup 87
\MT@define@set@key@: don't expand variables imme-

diately (requested by *Georg Verweyen*) 90
\MT@get@highlevel: no longer check whether defaults
    have changed 91
\MT@ifdefined@c@T: new macros: true case only . . 19
\MT@ifint: use \pdfmatch if available 20
\MT@ifstreq: use \pdfstrcmp if available 22
\MT@in@clist: fix 23
\MT@info@missing@char: info instead of warning
    (after *Michael Hoppe* reported that the 'fl' ligature
    is missing in Palatino SC) 45
\MT@is@feature: new macro: check for pdfTeX feature 26
\MT@map@clist@n: following LaTeX3 23
\MT@permute@@@@@: don't define permutations for
    unused encodings 107
\MT@rem@from@clist: fix 23
\MT@setup@: defer setup until the end of the preamble 27

2006/01/20 Version 1.9b

General: compatibility with listings: sanitise more
    catcodes (reported by *Holger Uhr*) 31
  compatibility with the extendedchar option of the
    listings package 31
Documentation: activate expansion in the distributed
    PDF U1

add samples of micro-typographic features U3
\MT@features: use throughout the package to adjust
    to beta-ness 26
\MT@ifdimen: use \pdfmatch if available 21
\MT@warn@code@too@large: fix calculation with
    present factor 45

2006/02/02 Version 1.9c

Documentation: add example of how to increase protrusion of footnote markers (suggested by *Georg Verweyen*) U21
Protrusion: settings for URW Garamond 142

\MT@define@code@key@font: fix: context was ignored 100
\MT@define@code@key@size: fix: embrace
    \MT@tempsize in \csname (bug introduced in
    v1.9b) 100

2006/05/05   **Version 1.9d**

2006/07/28   **Version 1.9e**

2006/09/09   **Version 1.9f**

2007/01/14   **Version 2.0**

\MT@split@name: adjust to possible letterspacing .. 38
\SetExtraKerning: new command: additional kerning 99
\SetExtraSpacing: new command: adjustment of interword spacing 99

\SetTracking: new command: tracking 98
\textls: new command: letterspacing 68
  starred version: remove spaces around text 68
\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations 7

2007/01/21 Version 2.1

General: compatibility with pinyin: disable microtype in \py@macron (reported by *Sven Naumann*) 31
  fix: letterspace package forgot to load keyval . 16
\lslig: new command: protect ligatures in letterspaced text 63

\MT@get@ls@basefont: redone: use \pdfmatch to make it bullet-proof 63
\MT@orig@pickupfont: compatibility with CJK: also check for its definition 84
\textls: fix: use \hmode@bgroup 68

2007/07/14 Version 2.2

General: disable microtype if wordcount is loaded (reported by *Ross Hetherington*) 27
  new option: copyfonts 112
  simplify key declarations 101
  use catcode trickery for e-TeX test 13
Documentation: add hint about error message with pdfTeX 1.40 .... U29
  add hint about extra TOC leader dot (first discovered by *Morten Høgholm*) U27
  add overview U4
  logo transparency and amusement U1
Font aliases: declare chr (chmath) as an alias of bch (reported by *Geoff Vallis*) 132
  declare fp9x, fp9j (FPL Neu) as aliases of ppl[xj] 131
Font sets: default set for tracking: smallcaps 130
Inheritance: remove '-' → '127' 134
Protrusion: settings for Bitstream Letter Gothic .. 142
Spacing: add sample 189
Tracking: add ligatures that are to be disabled ... 139
\DeclareMicrotypeVariants: new command 95
\DisableLigatures: new optional argument: disable selected ligatures only 97
\lslig: always defined 63
\MT@checklist@font: fix: construct font name from characteristics 41
\MT@copy@font: optionally work on copies of fonts . 36
\MT@get@basefamily: redone, working on font names and suffixes of arbitrary length 73
\MT@get@charwd: subtract letterspacing amount from width 44
\MT@get@ls@basefont: fix again: remember base font in a macro 63
\MT@ifdimen: employ LuaTeX features if available . 21
\MT@ifint: employ LuaTeX features if available ... 20
\MT@ifstreq: employ LuaTeX features if available . 22
  fix: ε-TeX version shouldn't use \x and \y (found by *Wiebke Petersen*) 22

\MT@is@composite: more robust: expand exactly once 82
\MT@is@symbol: expand once more (for frenchpro) 80
\MT@lsfont: use \font@name, not \MT@font 59
\MT@lua: (basic) support for LuaTeX 15
\MT@pdftex@no: case 7: pdfTeX 1.40.4 14
\MT@preset@aux@space: generalised 49
\MT@set@all@pr: (et al.) allow empty values 42
\MT@set@inputenc@: only load inputenc files if necessary 47
\MT@set@tr@codes: disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4) 60
  possibility to customise interword spacing 60
\MT@setup@expansion: warning if stretch or shrink aren't multiples of step 123
\MT@setupfont: don't call \@@enc@update anymore . 36
  only add features that are available with the respective pdfTeX 36
\MT@setupfont@hook: restore percent character if Galician babel is loaded 29
\MT@the@pr@code@tr: adjust protrusion of letterspaced fonts 42
\MT@tracking: remember fonts that shouldn't be letterspaced 58
\MT@tracking@: fix: tracking couldn't be re-enabled 58
\MT@warn@tracking@DVI: warning when letterspacing in DVI mode 126
\MT@with@babel@and@T: also inspect class options . 27
\pickup@font: letterspace: setup inside group ... 85
\SetTracking: new key 'no ligatures' to disable ligatures of letterspaced fonts 98
  new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by *Steven E. Harris*) 98
  third argument may be empty 98
\textmicrotypecontext: new command: wrapper around \microtypecontext 88

2007/12/23 Version 2.3

General: disable \microtypecontext in hyperref's \pdfstringdef 30
  fix: really switch off Turkish shorthands 128
  new value for verbose option: silent (suggested by *Karl Berry*) 113
  turned some warnings into errors 113

Documentation: add kerning sample U18
  add letterspacing illustration U16
\do@subst@correction: remember substitute font for all times (reported by *Stephan Hennig*) 86
\lslig: redone: extract outer kerns from current letterspacing amount 63

2010/01/10 Version 2.4

2013/03/13 Version 2.5

2013/05/23 Version 2.5a

2016/05/01 Version 2.6

2016/05/14 Version 2.6a

2017/07/07 Version 2.7

2018/01/14 Version 2.7a

2019/02/28 Version 2.7b

General: update lua function `microtype.info` after changes in `luaotfload` (reported by *Moritz Wemheuer* and *Ulrike Fischer*) 18
Documentation: update hint about non-7-bit characters (notified by *Frank Mittelbach*) U28
Inheritance: add textquotedblleft ligature to OT4

(reported by *Franz Wexler*) 135
`\MT@info@missing@char`: fix message for glyphs specified as names in XƎTEX (reported by *Paolo Ney*) 45
`\MT@setupfont`: always select current font with XƎTEX and LuaTEX (reported by *Paolo Ney*, solution by *Ulrike Fischer*) 35

2019/10/10 Version 2.7c

General: turn warning into info when overwriting the `keepligature` function (reported by *Andy N*) .. 70
`\MT@is@active`: compatibility with LATEX 2019/10/01 80

`\MT@is@symbol`: take care of `\remove@tlig` 80
`\showhyphens`: compatibility with LATEX 2019/10/01 (reported by *Phelype Oleinik* and *Falk Hanisch*) 124

2019/11/18 Version 2.7d

`\MT@copy@font@`: in LuaTEX, don't use the `\copyfont` primitive, but load the font anew (reported by *Paolo Polesana* and *Oliver Kopp*) 37
`\MT@register@subst@font`: remove substitute font

from list (reported by *Markus Kohm*) 87
`\MT@register@subst@font@cx`: remove substitute font from lists 87

2020/12/07 Version 2.8

General: `letterspace` works with e-TEX only 13
compatibility with `soul`: patch for font change (reported by *Md Ayquassar*) 31
fix for `luatexbase` 18
Documentation: declare `DVIoutput` option deprecated U8
squash fake news about automatic font expansion with `dvilualatex` U7
Font aliases: declare aliases for `step` and `domitian` (notified by *Daniel Benjamin Miller*) 131
declare aliases for `stix` and `stix2` fonts 131
declare New Computer Modern as an alias of Latin Modern Roman 131
Font sets: default set for expansion: `alltext-nott` (suggested by *Aman Mehra*) 130
default set for spacing: `alltext-nott` 130

`\lsstyle`: fix: enforce math setup, again 63
`\microtypecontext`: fix `activate` shortcut 88
ignore spaces 88
`\MT@do@font`: fix for XƎTEX 25
simplify lua function 25
`\MT@fix@fontdimen@six`: try to fix zero `\fontdimen 6` 38
`\MT@if@luaotf@font`: use lua function 24
`\MT@ifstreq`: use XƎTEX's `\stringcmp` 22
`\MT@setup@expansion`: warning when expanding in DVI mode with LuaTEX (reported by *Daniel Benjamin Miller*) 121
`\MT@tr@set@space@`: simplified 65
`\MT@tr@unit@`: fix: allow unit regardless whether letterspacing is set 62
`\textmicrotypecontext`: ignore spaces 88

2021/02/22 Version 2.8a

General: rename `mt-pad.cfg` to `mt-EBGaramond.cfg` (requested by *Karl Berry*) 129
rename `mt-PalatinoLinotype.cfg` to `mt-Palatino.cfg` (requested by *Karl Berry*) .. 194
Documentation: reorganise table 1 for clarification (suggested by *Daniel Benjamin Miller*) U6
Font aliases: add `-LF`, `-TLF`, `-OsF` and `-TOsF` as variants 130
declare `mlmodern` fonts as aliases of Latin Modern (reported by *Daniel Benjamin Miller*) 130
reference New Computer Modern also by file name (reported by *Canageek*) 131

Inheritance: specify 'ff' ligature as Unicode instead of glyph name 194
Protrusion: hide `euroitc` settings (requested by *Karl Berry*) 188
Spacing: add dummy settings (reported by *DORpapst*) 192
`\DeclareCharacterInheritance`: skip settings for tracking 104
`\MT@get@size@@`: compatibility with `svjour3` (reported by *Ekkehart Schlicht* (no relation!)) 93
`\MT@glet`: use LuaTEX's `\glet`, if available 19

2021/02/25 Version 2.8b

Inheritance: dummy settings for the Font Awesome font (`mt-FontAwesome.cfg`) 202

settings for the Lato font (`mt-Lato.cfg`) (reported by *dsedivec*) 202

2021/03/14 · Version 2.8c

2021/10/31 · Version 3.0

2021/12/02 · Version 3.0a

D Index

Numbers in upright shape refer to the *page* where the corresponding entry is described (bold face) resp. occurs. Numbers in italics refer to the *code line* where the corresponding entry is defined (underlined) resp. used. Numbers prefixed with 'U' refer to the User manual.

E   The LATEX Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 LATEX3 Project

Preamble

The LATEX Project Public License (LPPL) is the primary license under which the LATEX kernel and the base LATEX packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is TEX-related (such as a LATEX package), but it is written in such a way that you can use it even if your work is unrelated to TEX.

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

The section 'WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE', below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the LATEX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document 'cfgguide.tex' and 'modguide.tex' in the base LATEX distribution for suggestions.

Definitions

In this license document the following terms are used:

Work: Any work being distributed under this License.

Derived Work: Any work that under any applicable law is derived from the Work.

Modification: Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify: To apply any procedure that produces a Derived Work under any applicable law.

Distribution: Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun's Network File System (NFS).

Compiled Work: A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer: A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the 'Copyright Holder' under any applicable law.

Base Interpreter: A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a 'LATEX-Format' or in the case of files belonging to the 'LATEX-format' a program implementing the 'TEX language'.

Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.

2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.

3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.

4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.

5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.

6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.

   (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.

   (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

   (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide

any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.

(d) You distribute at least one of the following with the Derived Work:

i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;

ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.

7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.

8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.

9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.

10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.

(b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.

12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work 'as is', without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status 'author-maintained' if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is 'author-maintained'.

The Work has the status 'maintained' if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status 'maintained' to 'unmaintained' if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.

2. If this search is successful, then enquire whether the Work is still maintained.

(a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.

(b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current

Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a LATEX work, this could be done, for example, by posting to comp.text.tex.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.

   (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.

4. If you make an 'intention announcement' as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then

you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as 'you' in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document 'modguide.tex' in the base LATEX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing LATEX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to LATEX, the discussion in 'modguide.tex' may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%    https://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status `maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the 'Work' referring to the three files 'pig.dtx', 'pig.ins', and 'pig.sty' (the last being generated from 'pig.dtx' using 'pig.ins'), the 'Base Interpreter' referring to any 'LATEX-Format', and both 'Copyright Holder' and 'Current Maintainer' referring to the person 'M. Y. Name'.

If you do not want the Maintenance section of LPPL to apply to your Work, change 'maintained' above into 'author-maintained'. However, we recommend that you use 'maintained' as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

Important Recommendations

Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.