

The microtype package

Subliminal refinements towards typographical perfection

— IMPLEMENTATION —

R Schlicht

w.m.l@gmx.net

v3.1b

2024/03/29

<https://github.com/schlicht/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 `XYTeX`: 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 `XYTeX` (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).

This package is copyright © 2004–2024 R Schlicht. It may be distributed and/or modified under the conditions of the [\$\LaTeX\$ Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘maintained’.

Contents

1	Implementation	4
1.1	Preliminaries	5
	Debugging [6] Visual debugging [8] Requirements [12] Declarations [17]	
	Auxiliary macros [18] Compatibility [27] Protrusion patches [32]	
1.2	Font setup	37
	Protrusion [42] Manual protrusion [51] Expansion [59] Interword spacing	
	(glue) [62] Additional kerning [63] Tracking [64] Disabling ligatures [75]	
	Loading the configuration [77] Translating characters into slots [81] Hook into	
	L ^A T _E X's font selection [90] Context-sensitive setup [94]	
1.3	Configuration	96
	Font sets [96] Variants and aliases [102] Configuration file management [103]	
	Disabling ligatures [104] Interaction with babel [104] Fine tuning [104] Character	
	inheritance [111] Permutation [113]	
1.4	Package options	117
	Declaring the options [117] Loading the definition file [121] Reading the	
	configuration file [122] Hook for other packages [123] Changing options	
	later [124] Processing the options [126]	
2	Configuration files	137
2.1	Font sets	137
2.2	Font variants and aliases	138
2.3	Interaction with babel	141
2.4	Note on admissible characters	141
2.5	Character inheritance	141
	OT1 [142] T1 [142] LY1 [143] OT4 [143] QX [144] T5 [144] EU1, EU2, TU [145]	
	LGR [146] Euro symbols [146]	
2.6	Tracking	147
2.7	Font expansion	147
2.8	Character protrusion	149
	Normal [150] Italics [160] Small caps [173] Italic small caps [177] Text	
	companion [180] Computer Modern math [184] AMS symbols [188] Euler [191]	
	Euro symbols [196]	
2.9	Interword spacing	196
	Nonfrenchspacing [198]	
2.10	Additional kerning	200
	French [200] Turkish [201]	
3	OpenType configuration files	202
3.1	Character inheritance	202
	Latin Modern Roman/New Computer Modern [202] Charis SIL [204] EB Gara-	
	mond [206] Palatino [209] Basic glyph set [210] Empty glyph set [210]	
3.2	Character protrusion	211
	Latin Modern Roman/New Computer Modern [211] Charis SIL [219] EB Gara-	
	mond [224] Palatino [228] Basic glyph set [231] Empty glyph set [231]	
4	Auxiliary file for micro fine tuning	233

A	The title logo	234
A.1	Macros	234
A.2	Document	239
B	The letterspacing illustration	239
B.1	Macros	239
B.2	Document	242
C	Change history	244
D	Index	256
E	The L^AT_EX Project Public License	268

List of Tables

1	Order for matching font attributes	79
---	--	----

List of Figures

1	Example of interword spacing	197
---	--	-----

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 XeTeX (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
                  censo: 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*\DeclareMicrotypeFilePrefix[1] {}
46 \newcommand*\DeclareMicrotypeBabelHook[2] {}
47 \newcommand*\microtypesetup[1] {}
48 \newcommand*\microtypecontext[1] {}
49 \newcommand*\textmicrotypecontext[2] {#2}
50 \newcommand\leftprotrusion[1] {#1}
51 \newcommand\rightprotrusion[1] {#1}
52 \providecommand*\noprotrusion{}
53 \newcommand*\noprotrusionifmode{}

```

```

54 \ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
55 /package
56 \newcommand*{lsstyle}{}
57 \newcommand{textls}[2]{}{}
58 \def\textls#1#{}
59 \newcommand*{lslig}[1]{#1}
60 *package
61 }

```

These commands also have a starred version.

```

62 \def\DeclareMicrotypeSet#1#\@gobbletwo
63 \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.

```

64 \@onlypreamble\DeclareMicrotypeSet
65 \@onlypreamble\UseMicrotypeSet
66 \@onlypreamble\DeclareMicrotypeSetDefault
67 \@onlypreamble\DisableLigatures
68 \@onlypreamble\DeclareMicrotypeVariants
69 \@onlypreamble\DeclareMicrotypeBabelHook
70 \@onlypreamble\DeclareMicrotypeFilePrefix

```

Don't load letterspace.

```

71 \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.

```

72 \def\MT@old@cmd#1#2{%
73   \newcommand*#1{\MT@error{%
74     \string#1 is deprecated. Please use\MessageBreak
75     \string#2 instead}{As I said}%
76   \let #1#2}}
77 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
78 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
79 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
80 \MT@old@cmd\LoadMicroTypeFile \LoadMicrotypeFile
81 /package

```

`\MT@warning` Communicate.

```

\MT@warning@n1 82 \def\MT@warning{\PackageWarning\MT@MT}
\MT@info 83 \def\MT@warning@n1#1{\MT@warning{#1\@gobble}}
\MT@info@n1 84 *package
\MT@vinfo 85 \def\MT@info{\PackageInfo\MT@MT}
\MT@vinfo 86 \def\MT@info@n1#1{\MT@info{#1\@gobble}}
\MT@error 87 \let\MT@vinfo\@gobble
\MT@warn@err 88 \def\MT@error{\PackageError\MT@MT}
89 \def\MT@warn@err#1{\MT@error{#1}{%
90   This error message appears because you loaded the `~\MT@MT'\MessageBreak
91   package with the option `verbose=errors'. Consult the documentation\MessageBreak
92   in \MT@MT.pdf to find out what went wrong.}}

```

1.1.1 Debugging

`\tracingmicrotype` Cases for `\tracingmicrotype`:

```

\MT@dinfo 0: almost none
\MT@dinfo@n1 1: + sets & lists
2: + heirs

```

3: + slots

4: + factors

```

93 (+debug)
94 \MT@warning@n1{This is the debug version}
95 \newcount\tracingmicrotype
96 \tracingmicrotype=2
97 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
98 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1}@gobble}\MT@addto@annot{#1}}
99 \let\MT@vinfo\MT@info@n1
100 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
101 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1}@gobble}\MT@addto@annot{Warning: #1}}
102 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
103 \def\MT@dinfo@n1#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@n1{#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

```
104 \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
\MT@addto@annot
\ifMT@inannot

```

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

```

105 \RequirePackage{pdftexcmds}
106 \newif\ifMT@inannot \MT@inannottrue
107 \let\MT@pdf@annot\@empty
108 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
109   {\def\MessageBreak{^^J@spaces}%
110   \MT@xadd\MT@pdf@annot{\pdf@escapestring{#1^^J}}}\fi\fi}

```

\iftracingmicrotypeinpdfall

With \tracingmicrotypeinpdfall false, 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.

```
111 \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).

```

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

```

```

126 }%
127 \iftracingmicrotypeinpdfall\kern1pt \fi
128 \global\MT@inannotfalse
129 \fi
130 }
131 </debug>
132 </package>
133 </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 1.2.8). First, some preparation:

```

134 < *show >
135 \RequirePackage{iftex}
136 \ifetex\else
137 \PackageError{microtype-show}
138 {This package only works with e-TeX}{Use e-TeX}
139 \fi
140 \ifxetex
141 \PackageError{microtype-show}
142 {This package only works with pdfTeX or luaTeX}{Don't use XeTeX}
143 \fi
144 \PackageWarning{microtype-show}{DO NOT USE THIS PACKAGE FOR REAL DOCUMENTS\@gobble}
145 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{microtype}}
146 \ProcessOptions\relax
147 \PassOptionsToPackage{verbose}{microtype}
148 \RequirePackage{microtype,graphicx,xcolor}

```

`\ifShowGlyphIndex` The following commands are configurable:

```

\ifShowMissingGlyphs 149 \newif\ifShowGlyphIndex
\GlyphScaleFactor 150 \newif\ifShowMissingGlyphs
\Showbaselinecolor 151 \newcommand*\GlyphScaleFactor{2}
\Showposcolor 152 \newcommand*\Showbaselinecolor{\color{black!40}}
\Shownegcolor 153 \newcommand*\Showposcolor{\color{green!50}}
154 \newcommand*\Shownegcolor{\color{red!50}}
\MTS@printtext Make sure to have a readable font.
\MTS@show@index 155 \ifluatex
\MTS@crulefill 156 \def\MTS@printtext#1{\usefont{TU}{lmr}{m}{n}#1}
157 \else
158 \def\MTS@printtext#1{\usefont{T1}{cmr}{m}{n}#1}
159 \fi
160 \def\MTS@show@index#1{\ifShowGlyphIndex{\tiny$_{#1}$%
161 % \ifluatex^{\mathrm{#1}}%
162 % \MT@lua{tex.print(luaotfload.aux.name_of_slot(tonumber([[#1]]))}}}\fi
163 % }\fi\space}
164 \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 165 \g@addto@macro\MT@setupfont{\MTS@Prot\MTS@Char}
166 \let\MTS@Prot\relax
167 \let\MTS@Char\relax

```

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

```

\MTS@glyphlist 168 \def\MTS@setup{%
169 \fboxsep=0pt
170 \fboxrule=.1pt
171 \raggedright
172 \let\MTS@glyphlist\@gobble
173 \def\MT@feat{pr}%
174 }

```


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

```
175 \newcommand*\ShowProtrusion{%
176   \begingroup
177   \MTS@setup
178   \let\MTS@Prot\MTS@Prot@do
179   \def\MT@cat{c}%
180   \selectfont
181 }
```

`\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.

```
182 \def\MTS@Prot@do{%
183   \MT@ltx@pickupfont
184   \let\MT@pr@split@val\MTS@pr@split@val
185   \let\MT@load@list\MTS@load@list
186   \let\MT@set@pr@prefixes@\MTS@set@pr@prefixes@
187   \MTS@show@pr
188   \endgroup
189   \aftergroup\selectfont
190 }
```

`\ShowCharacterInheritance`

```
191 \newcommand*\ShowCharacterInheritance{%
192   \begingroup
193   \MTS@setup
194   \let\MTS@Char\MTS@Char@do
195   \def\MT@cat{inh}%
196   \selectfont
197 }
```

`\MTS@Char@do`

```
198 \def\MTS@Char@do{%
199   \MT@ltx@pickupfont
200   \let\MT@set@pr@prefixes@\MTS@set@pr@prefixes@
201   \MTS@show@inheritance
202   \endgroup
203   \aftergroup\selectfont
204 }
```

`\ShowProtrusionLineGlyph` By glyph.

```
205 \newcommand*\ShowProtrusionLineGlyph[1]{%
206   {\MTS@setup
207   \MTS@showprotrusionline{~#1}}%
208 }
```

`\ShowProtrusionLineIndex` By glyph number.

```
209 \newcommand*\ShowProtrusionLineIndex[1]{%
210   {\MTS@setup
211   \MTS@showprotrusionline{#1}}%
212 }
```

`\MTS@showprotrusionline`

```
\MTS@lpcode 213 \def\MTS@showprotrusionline#1{%
\MTS@rprcode 214   \edef\MTS@lpcode{\number\lpcode\font#1}%
215   \edef\MTS@rprcode{\number\rprcode\font#1}%
216   \char#1%
217   lorem ipsum dolor sit amet, \MTS@crulefill\ %
218   \MTS@printtext{\ifnum\MTS@lpcode=z@Showbaselinecolor\fi[\MTS@lpcode]}
219   \fbox{\char#1}\MTS@show@index{\number#1}
220   \MTS@printtext{\ifnum\MTS@rprcode=z@Showbaselinecolor\fi[\MTS@rprcode]}
221   \MTS@crulefill\ you know the rest%
222   \char#1\par
223   \ShowDummyLine
224 }
```

`\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.

```
225 \newcommand*\ShowDummyLine{%
226 {\fontencoding{\encodingdefault}\fontseries{\seriesdefault}\fontshape{\shapedefault}%
227 \selectfont\noindent
228 here is the beginning of a line, \dotfill and here is its end}\par
229 }
```

`\ShowProtrusionAll`

```
230 \newcommand*\ShowProtrusionAll{%
231 {\MTS@setup
232 \MTS@lede{}}%
233 \MT@do@font{\iffontchar\font\@tempcnta\MTS@showprotrusionline{\@tempcnta}\fi}}%
234 }
```

`\ShowProtrusionDefined`

```
235 \newcommand*\ShowProtrusionDefined{%
236 {\MTS@setup
237 \MTS@lede{defined}%
238 \let\MTS@first@gobble
239 \let\MTS@second@firstofone
240 \MT@do@font{%
241 \MTS@firstorsecond
242 \MTS@temp{%
243 \iffontchar\font\@tempcnta\MTS@showprotrusionline{\@tempcnta}\else
244 \MT@warning@n1{Glyph \the\@tempcnta\space is missing in font
245 \MessageBreak\font@name}%
246 \fi}}}%
247 }
```

`\ShowProtrusionMissing`

```
248 \newcommand*\ShowProtrusionMissing{%
249 {\MTS@setup
250 \MTS@lede{missing}%
251 \let\MTS@first@firstofone
252 \let\MTS@second@gobble
253 \MT@do@font{%
254 \MTS@firstorsecond
255 \iffontchar\font\@tempcnta\MTS@temp{\MTS@showprotrusionline{\@tempcnta}}\fi}}%
256 }
```

`\MTS@lede`

```
257 \def\MTS@lede#1{%
258 \selectfont
259 \edef\MTS@font{\expandafter\string\font@name}%
260 \MTS@printtext{All glyphs \MT@ifempty{#1}{in}{#1 in protrusion list for}
261 font \texttt{\MTS@font:}\par
262 \ShowDummyLine
263 }
```

`\MTS@firstorsecond`

```
264 \def\MTS@firstorsecond{%
265 \let\MTS@temp\MTS@first
266 \ifnum\lcode\font\@tempcnta=\z@ \else
267 \let\MTS@temp\MTS@second
268 \fi
269 \ifnum\rpcode\font\@tempcnta=\z@ \else
270 \let\MTS@temp\MTS@second
271 \fi
272 }
```

`\MTS@charwd` Display the glyph with protrusion.

`\MTS@lp@` 273 \newdimen\MTS@charwd

`\MTS@rp@`

`\MTS@show@char@pr`

```

274 \newdimen\MTS@lp@
275 \newdimen\MTS@rp@
276 \def\MTS@show@char@pr#1{%
277   \xdef\MTS@glyphlist{\MTS@glyphlist,#1}%
278   \scalebox{\GlyphScaleFactor}{\strut\escapechar~\
279     \MTS@charwd=\fontcharwd\MT@font#1\relax

```

The baseline rule.

```

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

```

Left protrusion.

```

282   {\ifdim\MTS@lp@<\z@\Shownegcolor\else\Showposcolor\fi
283     \vrule width \ifdim\MTS@lp@<\z@ -\fi\MTS@lp@ height 1em depth .2em}%
284   \hskip\dimexpr\MTS@charwd\ifdim\MTS@lp@>\z@-\MTS@lp@\fi
285     \ifdim\MTS@rp@>\z@-\MTS@rp@\fi\relax

```

Right protrusion.

```

286   {\ifdim\MTS@rp@<\z@\Shownegcolor\else\Showposcolor\fi
287     \vrule width \ifdim\MTS@rp@<\z@ -\fi\MTS@rp@ height 1em depth .2em}%
288   \hskip-\dimexpr\MTS@charwd+\fbxrule\ifdim\MTS@rp@<\z@-\MTS@rp@\fi\relax

```

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

```

289   \fbox{\char#1}}\,%
290   \MTS@show@index{#1}%
291 }

```

`\MTS@show@char` Just show the glyph; the second command also remembers it.

```

\MTS@show@char@x 292 \def\MTS@show@char#1{\scalebox{\GlyphScaleFactor}{%
293   \strut\fbox{\char#1}}\MTS@show@index{#1}}
294 \def\MTS@show@char@x#1{\xdef\MTS@glyphlist{\MTS@glyphlist,#1}\MTS@show@char{#1}}

```

`\MTS@show@missing`

```

295 \def\MTS@show@missing{%
296   \MT@ifdefined@c@T\MT@pr@inh@name{%
297     \MTS@lp@=\z@ \MTS@rp@=\z@
298     \par \MTS@printtext{Glyphs not included in configuration (with defined heirs):}%
299     \MT@do@font{%
300       \edef\MT@temp{\the\@tempcnta}%
301       \MT@ifdefined@n@T\MT@inh@\MT@pr@inh@name @\MT@temp @}{%
302         \MT@exp@one@n\MT@in@clist\MT@temp\MTS@glyphlist
303         \ifMT@inlist@else \newline
304         \llap{\MTS@show@char@pr{\MT@temp} \MTS@printtext{=} }%
305         \MT@exp@cs\MT@map@tlist@c
306         {MT@inh@\MT@pr@inh@name @\the\@tempcnta @}%
307         \MTS@show@char@x
308       \fi
309     }%
310   }%
311 }%
312 \MTS@show@missing@
313 }

```

`\MTS@show@missing@`

```

314 \def\MTS@show@missing@{%
315   \par \MTS@printtext{Other glyphs not in configuration:}\newline
316   \MT@do@font{%
317     \edef\MT@temp{\the\@tempcnta}%
318     \MT@exp@one@n\MT@in@clist\MT@temp\MTS@glyphlist
319     \ifMT@inlist@else
320       \MTS@show@char\MT@temp
321     \fi
322   }%
323 }

```

`\MTS@show@inheritance`

```

324 \def\MTS@show@inheritance{%
325   \MT@get@inh@list
326   \MTS@printtext{Character inheritance for font `~\texttt{\MT@font}' : \\
327   \MT@ifdefined@c@TF\MT@listname{%
328     \MTS@printtext{First matching list is for `~\texttt{\@tempa}' : \\
329     \texttt{\MT@listname} : } \par \leavevmode
330   \MT@do@font{%
331     \MT@ifdefined@n@T{MT@inh@MT@listname @\the\@tempcnta @}{%
332       \newline
333       \xdef\MTS@glyphlist{\MTS@glyphlist,\the\@tempcnta}%
334       \llap{\MTS@show@char{\the\@tempcnta}\MTS@printtext{= }}%
335       \MT@exp@cs\MT@map@tlist@c
336       {MT@inh@MT@listname @\the\@tempcnta @}%
337       \MTS@show@char@x
338     }%
339   }%
340   \MT@ifdefined@n@T{MT@inh@MT@listname @prefixes}{%
341     \par \MTS@printtext{(with prefixes :)}%
342     \@tempcntb=\z@
343     \let\MTS@show@char@pr\MTS@show@char@x
344     \MT@set@pr@prefixheirs}%
345   \ifShowMissingGlyphs\MTS@show@missing@f{i
346 }%
347   \MTS@printtext{NOT DEFINED}%
348 }%
349 \par
350 }
351 </show>

```

1.1.3 Requirements

Back to the user packages.

`\MT@plain` The letterspace package works with:

- 0: miniltx
- 1: eplain
- 2: L^AT_EX

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

```

352 <*package|letterspace>
353 <*plain>
354 \def\MT@plain{2}
355 \ifx\documentclass@undefined
356   \def\MT@plain{1}
357   \def\hmode@bgroup{\leavevmode\bgroup}
358   \def\nfss@text#1{{\mbox{#1}}}
359   \let@typeset@protect\relax
360   \ifx\epain@undefined
361     \def\MT@plain{0}
362     \def\PackageWarning#1#2{%
363       \begingroup
364         \newlinechar=10 %
365         \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces}%
366         \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
367       \endgroup
368     }
369     \def\on@line{ on input line \the\inputlineno}
370     \def\@spaces{\space\space\space\space}
371   \fi
372 \fi

```

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

```
373 \def\MT@requires@latex#1{%
374   \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
375 }
376 </plain>
```

For definitions that depend on e-TeX features.

```
377 \ifcase 0%
378   \ifx\TeXversion\undefined 1\else
379   \ifx\TeXversion\relax 1\else
380   \ifcase\TeXversion 1\fi
381   \fi
382 \fi
383 \else
384   \catcode`\^^Q=9 \catcode`\^^X=14
385 \fi
386 <letterspace>^^Q\MT@warning@n1{This package requires the etex extensions.
387 <letterspace>^^Q \MessageBreak Exiting}\MT@restore@catcodes\endinput
388 <debug>\MT@dinfo@n1{0}{this is
389 <debug>^^Q not
390 <debug> etex}
```

We check whether we are running pdfTeX, XeTeX, 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.

```
391 \def\MT@clear@options{%
392 <plain> \MT@requires@latex1{%
393   \AtEndOfPackage{\let\unprocessedoptions\relax\MT@restore@catcodes}%
394   \let\CurrentOption\@empty
395 <plain> }\relax
396 }
```

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.

```
397 \ifx\normalpdftexversion\undefined \else
398   \let\pdftexversion \normalpdftexversion
399   \let\pdftexrevision \normalpdftexrevision
400   \let\pdfoutput \normalpdfoutput
401 \fi
```

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

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

```

422 \ifx\directlua\@undefined \else
423   \ifx\directlua\relax \else
424     \def\MT@engine{lua}
425     \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.

```

426 <*\letterspace>
427   \let\MT@pdf@or@lua@secondoftwo
428   \ifnum\luatexversion < 62 \MT@engine@unfittrue
429   \else
430     \let\MT@lua\directlua
431     \ifnum\luatexversion > 84
432       \let\pdfoutput\outputmode
433       \let\pdfprotrudechars\protrudechars
434       \let\pdfadjustspacing\adjustspacing
435     \fi
436   \fi
437 </\letterspace>
438 \fi
439 \fi
440 <*\package>
441 \ifx\MT@engine\relax
442   \ifx\XeTeXversion\@undefined \else
443     \ifx\XeTeXversion\relax \else
444       \def\MT@engine{xe}
445       \def\MT@engine@minversion{0.9997}
446       \ifdim 0\XeTeXrevision pt > 0.9996pt
447         \MT@engine@unfitfalse
448       \fi
449     \fi
450   \fi
451 \fi
452 </\package>
453 </\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 ($\geq 0.14h$)
- 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`; `\pdfmatch1`; `\pdftracingfonts`; always e-TeX (≥ 1.40)
- 7: + `\letterspacefont` doesn't disable ligatures and kerns; `\pdfcopyfont` ($\geq 1.40.4$)
- 8: + `\letterspacefont` uses explicit `\fontdimen 6` if specified ($\geq 1.40.23$)

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

```

454 {*pdf-}
455 {debug}\MT@info@n1{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
456 \def\MT@pdftex@no{8}
457 \ifnum\pdftexversion = 140
458   \ifnum\pdftexrevision < 23
459     \def\MT@pdftex@no{7}
460     \ifnum\pdftexrevision < 4
461       \def\MT@pdftex@no{6}
462     \fi
463   \fi
464 \else
465   \ifnum\pdftexversion < 140
466     \def\MT@pdftex@no{5}
467     \ifnum\pdftexversion < 130
468       \def\MT@pdftex@no{4}
469       \ifnum\pdftexversion < 120
470         \def\MT@pdftex@no{3}
471         \ifnum\pdftexversion = 14
472           \ifnum \expandafter`\pdftexrevision < `h
473             \def\MT@pdftex@no{2}
474           \fi
475         \fi
476       \fi
477     \fi
478   \fi
479 \fi
480 {debug}\MT@info@n1{0}{pdftex no.: \MT@pdftex@no}
481 {/pdf-}

```

`\MT@xetex@no` X_YTeX 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 X_YTeX in the future.

```

482 {*xe-}
483 {debug}\MT@info@n1{0}{this is xetex (\the\XeTeXversion\XeTeXrevision)}
484 %\ifdim 0\XeTeXrevision pt < 0.9997pt
485 % \def\MT@xetex@no{1}
486 %\else
487 % \def\MT@xetex@no{2}
488 %\fi
489 {debug}%\MT@info@n1{0}{xetex no.: \MT@xetex@no}
490 {/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 glyphs² (≥ 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).

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).

```

491 (*lua-
492 debug)MT@info@n10{this is luatex (\the\luatexversion)}
\MT@lua    Communicate with lua. Beginning with LuaTeX 0.36, \directlua no longer requires
           a state number.
493 \let\MT@lua\directlua
494 \def\MT@luatex@no{6}
495 \ifnum\luatexversion<110
496   \def\MT@luatex@no{5}
497   \ifnum\luatexversion<90
498     \def\MT@luatex@no{4}
499     \ifnum\luatexversion<85
500       \def\MT@luatex@no{3}
501       \ifnum\luatexversion<62
502         \def\MT@luatex@no{2}
503         \ifnum\luatexversion<36
504           \def\MT@lua{\directlua0}
505           \def\MT@luatex@no{1}
506         \fi
507       \fi
508     \fi
509   \fi
510 \fi
511 debug)MT@info@n1{0}{luatex no.: \MT@luatex@no}
512 /lua-

```

Abort if no capable engine found.

```

513 (*package|letterspace)
514 \ifMT@engine@unfit
515   \MT@warning@n1{You
516     \ifx\MT@engine\relax
517       don't seem to be using pdftex%
518 (package)       , luatex or xetex%
519 (letterspace)   \space or luatex%
520     .\MessageBreak `MT@MT' only works with these engines.%
521   \else
522     are using a \MT@engine tex version older than
523 (package)       \MT@engine@minversion
524 (letterspace)   \MT@pdf@or@lua{1.40}{0.62}%
525     .\MessageBreak `MT@MT' does not work with this version.%
526     \MessageBreak Please install a newer version of \MT@engine tex.%
527   \fi
528   \MessageBreak I will quit now}
529 \MT@clear@options
530 \endinput\fi
531 /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.

```

532 (*package|letterspace)
533 \RequirePackage{keyval}[1997/11/10]
534 (*package)
535 ^^X\RequirePackage{etoolbox}
536 \providecommand\IfFormatAtLeastTF{\ifl@t@r\fmtversion}
\MT@toks    We need a token register,
537 \newtoks\MT@toks
\MT@tempbox  our own box,
538 \newbox\MT@tempbox
\ifMT@if@    and a scratch if.

```


539 \newif\ifMT@if@

1.1.4 Declarations

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

\ifMT@expansion 540 \newif\ifMT@protrusion
 \ifMT@auto 541 \newif\ifMT@expansion
 \ifMT@selected 542 \newif\ifMT@auto
 \ifMT@selected 543 \newif\ifMT@selected
 \ifMT@noligatures 544 \newif\ifMT@noligatures
 \ifMT@draft 545 \newif\ifMT@draft
 \ifMT@draft 546 \newif\ifMT@draft
 \ifMT@disable 547 \newif\ifMT@spacing
 \ifMT@spacing 548 \newif\ifMT@kerning
 \ifMT@kerning 549 \newif\ifMT@tracking
 \ifMT@tracking 550 \newif\ifMT@babel

 \ifMT@babel [This line intentionally left blank.]
 \MT@pr@level ... and numbers.
 \MT@ex@level 551 \let\MT@pr@level\@tw@
 \MT@pr@factor 552 \let\MT@ex@level\@tw@
 \MT@ex@factor 553 \let\MT@pr@factor\@m@
 \MT@ex@factor 554 \let\MT@ex@factor\@m@
 \MT@sp@factor 555 \let\MT@sp@factor\@m@
 \MT@kn@factor 556 \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 557 \let\MT@pr@unit\@empty
 558 \let\MT@sp@unit\@m@ne
 559 \def\MT@kn@unit{1em}

\MT@stretch Expansion settings.
 \MT@shrink 560 \let\MT@stretch\@m@ne
 \MT@step 561 \let\MT@shrink \@m@ne
 562 \let\MT@step \@m@ne

\MT@pr@min Minimum and maximum values allowed by pdfTeX.
 \MT@pr@max 563 \def\MT@pr@min{-\@m@}
 \MT@ex@min 564 \let\MT@pr@max\@m@
 \MT@ex@min 565 \let\MT@ex@min\z@
 \MT@ex@max 566 \let\MT@ex@max\@m@
 \MT@sp@min 567 \def\MT@sp@min{-\@m@}
 \MT@sp@max 568 \let\MT@sp@max\@m@
 \MT@kn@min 569 \def\MT@kn@min{-\@m@}
 \MT@kn@min 570 \let\MT@kn@max\@m@
 \MT@kn@max 571 */package*
 \MT@tr@min 572 \def\MT@tr@min{-\@m@}
 \MT@tr@min 573 \let\MT@tr@max\@m@
 \MT@tr@max 574 **package*

\MT@factor@default Default factor.
 575 \def\MT@factor@default{1000 }

\MT@stretch@default Default values for expansion.
 \MT@shrink@default 576 \def\MT@stretch@default{20 }
 577 \def\MT@shrink@default{20 }

\MT@letterspace Default value for letterspacing (in thousandths of 1 em).
 \MT@letterspace@default 578 */package*
 579 \let\MT@letterspace\@m@ne
 580 \def\MT@letterspace@default{100}

```

581 <*package>
\ifMT@document Our private test whether we're still in the preamble.
582 \newif\ifMT@document
583 </package>
584 </package|letterspace>

```

1.1.5 Auxiliary macros

```

\MT@requires@pdftex For definitions that depend on a particular pdfTeX resp. LuaTeX version.
\MT@requires@luatex 585 <*pdf-|lua->
586 \def
587 <pdf-> \MT@requires@pdftex%
588 <lua-> \MT@requires@luatex%
589 #1{\ifnum
590 <pdf-> \MT@pdftex@no
591 <lua-> \MT@luatex@no
592 <#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}
593 <lua-&debug>\MT@requires@luatex4{\MT@lua{tex.enableprimitives('pdf',{'tracingfonts'})}}\relax
594 <pdf-&debug>\MT@requires@pdftex6{
595 <debug>\pdftracingfonts=1
596 <pdf-&debug>\relax
597 </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 L^AT_EX, we load the luatexbase package.

```
598 <lua->\IfFormatAtLeastTF{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.

```

599 <lua->\IfFormatAtLeastTF{2017/01/01}\relax{\RequirePackage{luaotfload}}
600 <letterspace>\MT@pdf@or@lua\relax{
601 <letterspace>\ifx\newluafunction\@undefined \input ltluatex \fi
602 <lua-|letterspace>\MT@lua{require("microtype")}
603 <letterspace>}

```

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

```

604 <*luafile>
605
606 function microtype.info(...)
607   luatexbase.module_info("microtype",...)
608 end
609
610 local find      = string.find
611 local match     = string.match
612 local tex_write = tex.write
613
614 local catpackage
615 if luatexbase.registernumber then
616   catpackage = luatexbase.registernumber("catcodetable@atletter") -- LaTeX
617 else
618   catpackage = luatexbase.catcodetables.CatcodeTableLaTeXAtLetter -- luatexbase
619 end
620 function microtype.sprint (...)
621   tex.sprint(catpackage, ...)
622 end
623

```

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`.

```
624 if not math.tointeger or not pcall(math.tointeger,0) then
625   math.mininteger=-0x4FFFFFFFFFFFFF
626   math.maxinteger=0x4FFFFFFFFFFFFF
627   local floor=math.floor
628   function math.tointeger(n)
629     local f=floor(n)
630     return f==n and f or nil
631   end
632 end
633
634 </luafile>
```

To be continued, but first back to primitives.

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

```
635 <lua->\MT@requires@luatex6{\let\MT@glet\glet}\relax
636 <*package|letterspace>
637 \def\MT@glet{\global\let}
```

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

```
638 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
639 <*package>
640 \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 641 \def\MT@def@n{\MT@exp@cs\def}
642 \def\MT@gdef@n{\MT@exp@gcs\gdef}
```

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

```
\MT@xdef@n 643 </package>
644 \def\MT@edef@n{\MT@exp@cs\edef}
645 <*package>
646 \def\MT@xdef@n{\MT@exp@gcs\xdef}
```

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

```
\MT@glet@nc 647 \def\MT@let@nc{\MT@exp@cs\let}
648 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}
```

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

```
649 </package>
650 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
651 <*package>
```

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

```
\MT@glet@nn 652 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
653 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}
```

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

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

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

```
655 </package>
656 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}
```

`\MT@exp@two@c` Expand the next two tokens after `<#1>` once.

```
657 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
658 <*package>
```

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

```
659 \def\MT@exp@two@n#1#2#3{%
660   \expandafter\expandafter\expandafter
```

```
661 #1\expandafter\expandafter\expandafter
662 {\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` decreases memory use substantially.

```
\MT@ifdefined@n@TF 663 \def\MT@ifdefined@c@T#1{%
664 ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
665 ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
666 }
667 </package>
668 \def\MT@ifdefined@c@TF#1{%
669 ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
670 <package>^^Q \ifx#1\@undefined
671 <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
672 }
673 \def\MT@ifdefined@n@T#1{%
674 ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
675 <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
676 <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
677 }
678 \def\MT@ifdefined@n@TF#1{%
679 ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
680 <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
681 <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
682 }
```

`\MT@if@expanding@F` The following voodoo is based on a trick by *Ulrich Schwarz*.³

```
\MT@if@expanding@F 683 \def\MT@if@expanding@F{\let\MT@if@expanding@F@\MT@if@expanding@F@\@firstofone}
684 \def\MT@if@expanding@F@#1#2#3{\relax\relax}
685 <*package>
```

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

```
686 \def\MT@detokenize@n#1{%
687 ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil
688 ^^Q \string#1%
689 }
690 \def\MT@detokenize@c#1{%
691 ^^X \MT@exp@one@n\MT@detokenize@n#1%
692 ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
693 }
694 \def\MT@rem@last@space#1 #2{#1%
695 \ifx\@nil#2\else \space
696 \expandafter\MT@rem@last@space\expandafter#2\fi
697 }
```

`\MT@ifempty` Test whether argument is empty.

```
698 </package>
699 \begingroup
700 \catcode`\%=12
701 \catcode`\&=14
702 \gdef\MT@ifempty#1{&
703 \if %#1&
704 \expandafter\@firstoftwo
705 \else
706 \expandafter\@secondoftwo
707 \fi
708 }
709 \endgroup
```

3 Cf. <https://tex.stackexchange.com/a/29188/7674>

```

710 <*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).
711 </package>
712 </package|letterspace>
713 <pdf->\MT@requires@pdftex6{
714 <letterspace>\MT@pdf@or@lua{
715 <*pdf-|letterspace>
716 \def\MT@ifint#1{%
717   \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
718   \expandafter\@secondoftwo
719   \else
720   \expandafter\@firstoftwo
721   \fi
722 }
723 }{
724 </pdf-|letterspace>
725 <*pdf-|xe-|letterspace>
726 \def\MT@ifint#1{%
727   \if!\ifnum9<1#1!\else?\fi
728   \expandafter\@firstoftwo
729   \else
730   \expandafter\@secondoftwo
731   \fi
732 }
733 </pdf-|xe-|letterspace>
734 <pdf-|letterspace>}
735 <lua->\def\MT@ifint#1{\csname\MT@lua{microtype.if_int}([\#1])\endcsname}
736 <*luafile>
737 local function if_int(s)
738   if find(s,"^-[0-9]+ *$") then
739     tex_write("@firstoftwo")
740   else
741     tex_write("@secondoftwo")
742   end
743 end
744 microtype.if_int = if_int
745
746 </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.)
747 <*pdf->
748 \MT@requires@pdftex6{
749 \def\MT@ifdimen#1{%
750   \ifcase\pdfmatch{^[0-9]+([.],[0-9]+)?|.[.],[0-9]+)%
751   (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
752   \expandafter\@secondoftwo
753   \else
754   \expandafter\@firstoftwo
755   \fi
756 }
757 }{
758 </pdf->
759 <*pdf-|xe->
760 \def\MT@ifdimen#1{%
761   \setbox\z@=\hbox{%
762     \MT@count=1#1\relax
763     \ifnum\MT@count=\@ne
764       \aftergroup\@secondoftwo
765     \else
766       \aftergroup\@firstoftwo

```

```

767   \fi
768   }%
769 }
770 </pdf-|xe- >
771 <pdf- >
772 <lua- >\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen([[#1]])}\endcsname}
773 <*luafile >
774 local function if_dimen(s)
775   if (find(s, "^-*[0-9]+(%a*) *$") or
776       find(s, "^-*[0-9]*[.][0-9]+(%a*) *$")) then
777     tex_write("@firstoftwo")
778   else
779     tex_write("@secondoftwo")
780   end
781 end
782 microtype.if_dimen = if_dimen
783
784 </luafile >
\MT@ifdim   Compare floating point numbers.
785 <*package >
786 \def\MT@ifdim#1#2#3{%
787   \ifdim #1\p@ #2 #3\p@
788     \expandafter\@firstoftwo
789   \else
790     \expandafter\@secondoftwo
791   \fi
792 }
793 </package >
\MT@ifstreq   Test whether two strings (fully expanded) are equal.
794 <*pdf-|xe- >
795 <pdf- >\MT@requires@pdftex5{
796 \def\MT@ifstreq#1#2{%
797   \ifnum
798 <pdf- >   \pdfstrcmp
799 <xe- >   \strcmp
800     {#1}{#2}=\z@
801     \expandafter\@firstoftwo
802   \else
803     \expandafter\@secondoftwo
804   \fi
805 }
806 </pdf-|xe- >
807 <*pdf- >
808 }{
809 \def\MT@ifstreq#1#2{%
810   \edef\MT@resa{#1}%
811   \edef\MT@resb{#2}%
812   \ifx\MT@resa\MT@resb
813     \expandafter\@firstoftwo
814   \else
815     \expandafter\@secondoftwo
816   \fi
817 }
818 }
819 </pdf- >
820 <lua- >\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq([[#1]],[[#2]])}\endcsname}
821 <*luafile >
822 local function if_str_eq(s1, s2)
823   if s1 == s2 then
824     tex_write("@firstoftwo")
825   else
826     tex_write("@secondoftwo")
827   end

```

```

828 end
829 microtype.if_str_eq = if_str_eq
830
831 /luafile

```

With this, we can now also check whether versions match (using the command from 1.4.2).

```

832 lua-\MT@check@MT@version
833 lua- {\MT@lua{tex.write(microtype.module['date'] .. ' v' .. microtype.module['version'])}}
834 lua- {\MT@MT.lua}

```

`\MT@xadd` Add item to a list.

```

835 *package
836 \def\MT@xadd#1#2{%
837   \ifx#1\relax
838     \xdef#1{#2}%
839   \else
840     \xdef#1{#1#2}%
841   \fi
842 }

```

`\MT@xaddb` Add item to the beginning.

```

843 \def\MT@xaddb#1#2{%
844   \ifx#1\relax
845     \xdef#1{#2}%
846   \else
847     \xdef#1{#2#1}%
848   \fi
849 }
850 /package

```

`\MT@map@clist@n` Run `<#2>` on all elements of the comma list `<#1>`. This and the following is modelled after L^AT_EX3 commands.

`\MT@map@clist@c`

```

\MT@map@clist@ 851 *package | letterspace
\MT@map@clist@c 852 \def\MT@map@clist@n#1#2{%
\MT@clist@function 853   \ifx\@empty#1\else
\MT@clist@break 854     \def\MT@clist@function##1{#2}%
855     \MT@map@clist@#1,\@nil,\@nnil
856   \fi
857 }

858 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
859 \def\MT@map@clist@c#1,%
860   \ifx\@nil#1%
861     \expandafter\MT@clist@break
862   \fi
863   \MT@clist@function{#1}%
864   \MT@map@clist@
865 }
866 \let\MT@clist@function\@gobble
867 \def\MT@clist@break#1\@nnil{}
868 *package

```

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

```

\MT@map@tlist@c 869 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@map@tlist@n 870 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 871 \def\MT@map@tlist@n#1#2{%
872   \ifx\@nnil#2\else
873     #1{#2}%
874     \expandafter\MT@map@tlist@
875     \expandafter#1%
876   \fi
877 }
878 \def\MT@tlist@break#1\@nnil{\fi}

```

`\ifMT@inlist@` Test whether item $\langle \#1 \rangle$ is in comma list $\langle \#2 \rangle$. Using `\pdfmatch` would be slower.

```
\MT@inclist 879 \newif\ifMT@inlist@
880 \def\MT@inclist#1#2{%
881   \def\MT@res@a#1,#1,##2##3\@nnil{%
882     \ifx##2\@empty
883       \MT@inlist@false
884     \else
885       \MT@inlist@true
886     \fi
887   }%
888   \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
889 }
```

`\MT@rem@from@clist` Remove item $\langle \#1 \rangle$ from comma list $\langle \#2 \rangle$. This is basically `\@removeelement` from `ltnctr1.dtx`. Using `\pdfmatch` and `\pdflastmatch` here would be really slow!

```
890 \def\MT@rem@from@clist#1#2{%
891   \def\MT@res@a#1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
892   \def\MT@res@b#1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
893   \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
894 }
```

`\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.

```
895 \def\MT@in@tlist#1#2{%
896   \MT@inlist@false
897   \def\MT@res@a{#1}%
898   \MT@map@tlist@c#2\MT@in@tlist@
899 }
900 \def\MT@in@tlist@#1{%
901   \edef\MT@res@b{#1}%
902   \ifx\MT@res@a\MT@res@b
903     \MT@inlist@true
904     \expandafter\MT@tlist@break
905   \fi
906 }
```

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

```
\MT@in@rlist@ \MT@size@name
\MT@in@rlist@@ 907 \def\MT@in@rlist#1{%
\MT@size@name 908   \MT@inlist@false
909   \MT@map@tlist@c#1\MT@in@rlist@
910 }
911 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
912 \def\MT@in@rlist@@#1#2#3{%
913   \MT@ifdim{#2}=\m@ne{%
914     \MT@ifdim{#1}=\MT@size
915     \MT@inlist@true
916     \relax
917   }%
918   \MT@ifdim\MT@size<{#1}\relax{%
919     \MT@ifdim\MT@size<{#2}%
920     \MT@inlist@true
921     \relax
922   }%
923 }%
924 \ifMT@inlist@
925   \def\MT@size@name{#3}%
926   \expandafter\MT@tlist@break
927 \fi
928 }
```

`\MT@loop` This is the same as L^AT_EX's `\loop`, which we mustn't use, since this could confuse an
`\MT@iterate`
`\MT@repeat`

outer `\loop` in the document.

```

929 </package>
930 \def\MT@loop#1\MT@repeat{%
931   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
932   \MT@iterate \let\MT@iterate\relax
933 }
934 \let\MT@repeat\fi

```

`\MT@while@num` Execute `<#3>` from `<#1>` up to (excluding) `<#2>` (much faster than L^AT_EX's `\@whilenum`).

```

935 \def\MT@while@num#1#2#3{%
936   \@tempcnta#1\relax
937   \MT@loop #3%
938   \advance\@tempcnta \@ne
939   \ifnum\@tempcnta < #2\MT@repeat
940 }
941 </package|letterspace>

```

`\MT@if@luaotf@font` For fonts loaded by `luaotfload` we query the font's table.

```

942 <letterspace>\MT@pdf@or@lua{\let\MT@if@luaotf@font\@secondoftwo}{
943 <*lua-|letterspace>
944 \def\MT@if@luaotf@font{\csname\MT@lua{%
945   microtype.if_luaotf_font()
946   }\endcsname
947 }
948 </lua-|letterspace>
949 <letterspace>}
950 <*luafile>
951 local function if_luaotf_font()
952   local thefont = font.getfont(font.current())
953   if thefont and ( thefont.format == "opentype" or thefont.format == "truetype" )
954     then tex.write("@firstoftwo")
955     else tex.write("@secondoftwo")
956   end
957 end
958 microtype.if_luaotf_font = if_luaotf_font
959
960 </luafile>

```

`\MT@do@font` Execute `<#1>` 256 times,

```

961 <pdf-|letterspace>\def\MT@do@font{\MT@while@num\z@\cc\vi}

```

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

```

962 <*lua->
963 \def\MT@do@font#1{%
964   \MT@if@luaotf@font{%
965     \def\MT@dofont@function{#1}%
966     \MT@lua{microtype.do_font()}%
967   }{\MT@while@num\z@\cc\vi{#1}}%
968 }
969 </lua->

```

This is the lua function, which is much faster than looping through all glyphs in T_EX. 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.

```

970 <*luafile>
971 local function do_font()
972   local thefont = font.getfont(font.current())
973   if thefont then
974     for i,v in next,thefont.characters do
975       if v.index == nil or ( v.index > 0 and i < 1114112 ) then
976         microtype.sprint([[@tempcnta=]]..i..[[\relax\MT@dofont@function]])

```

```

977     end
978   end
979 end
980 end
981 microtype.do_font = do_font
982
983 (/luafile)

```

The X_YTeX variant (it's slow ...!).

```

984 (*xe-)
985 \def\MT@do@font#1{%
986   \@tempcnta=\z@
987   \MT@loop
988   \iffontchar\MT@font\@tempcnta #1\fi
989   \advance\@tempcnta\@ne
990   \ifnum\@tempcnta < \XeTeXlastfontchar\MT@font \MT@repeat
991 }
992 (/xe-)
993 (*package)

```

`\MT@count` Increment macro `<#1>` by one. Saves using up too many counters. The e-TeX way is slightly faster.

```

994 \newcount\MT@count
995 \def\MT@increment#1{%
996   ^^X \edef#1{\number\numexpr #1 + 1\relax}%
997   ^^Q \MT@count=#1\relax
998   ^^Q \advance\MT@count \@ne
999   ^^Q \edef#1{\number\MT@count}%
1000 }

```

`\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.

```

1001 \def\MT@scale#1#2#3{%
1002   ^^Q \multiply #1 #2\relax
1003   \ifnum #3 = \z@
1004     ^^X #1=\numexpr #1 * #2\relax
1005   \else
1006     ^^X #1=\numexpr #1 * #2 / #3\relax
1007     ^^Q \divide #1 #3\relax
1008   \fi
1009 }

```

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

```

\MT@abbr@ex
\MT@abbr@pr@c 1010 \def\MT@abbr@pr{protrusion}
\MT@abbr@ex@c 1011 \def\MT@abbr@ex{expansion}
\MT@abbr@pr@inh 1012 \def\MT@abbr@pr@c{protrusion codes}
\MT@abbr@ex@inh 1013 \def\MT@abbr@ex@c{expansion codes}
\MT@abbr@n1 1014 \def\MT@abbr@pr@inh{protrusion inheritance}
\MT@abbr@sp 1015 \def\MT@abbr@ex@inh{expansion inheritance}
\MT@abbr@sp@c 1016 \def\MT@abbr@n1{no ligatures}
\MT@abbr@sp@inh 1017 \def\MT@abbr@sp{spacing}
\MT@abbr@kn 1018 \def\MT@abbr@sp@c{interword spacing codes}
\MT@abbr@kn@c 1019 \def\MT@abbr@sp@inh{interword spacing inheritance}
\MT@abbr@kn@inh 1020 \def\MT@abbr@kn{kerining}
\MT@abbr@tr 1021 \def\MT@abbr@kn@c{kerining codes}
\MT@abbr@tr@c 1022 \def\MT@abbr@kn@inh{kerining inheritance}
\MT@abbr@tr@inh 1023 \def\MT@abbr@tr{tracking}
\MT@abbr@tr@c 1024 \def\MT@abbr@tr@c{tracking amount}

```

These we also need the other way round.

```

\MT@rba@protrusion
\MT@rba@expansion
\MT@rba@spacing
\MT@rba@kerining
\MT@rba@tracking

```

```

1025 \def\MT@rbba@protrusion{pr}
1026 \def\MT@rbba@expansion{ex}
1027 \def\MT@rbba@spacing{sp}
1028 \def\MT@rbba@kerning{kn}
1029 \def\MT@rbba@tracking{tr}

\MT@features    We can work on these lists to save some guards in the dtx file.
\MT@features@long 1030 \def\MT@features{pr,ex,sp,kn,tr}
                  1031 \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.
1032 \def\MT@is@feature#1#2{%
1033   \MT@in@clist{#1}\MT@features@long
1034   \ifMT@inlist@
1035     \expandafter\@firstofone
1036   \else
1037     \MT@error{`#1' is not an available micro-typographic\MessageBreak
1038       feature. Ignoring #2}{Available features are: `~\MT@features@long'.}%
1039     \expandafter\@gobble
1040   \fi
1041 }

```

1.1.6 Compatibility

For the record, the following L^AT_EX kernel commands will be modified by `microtype`:

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent` (all in section 1.2.10)
- `\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.

```

1042 \@ifl@aded{tex}{wordcount}{%
1043   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
1044     Disabling `~\MT@MT', since it wouldn't work}%
1045   \MT@clear@options@endinput}\relax

```

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

```

1046 \@ifclassloaded{minimal}{%
1047   \MT@warning@nl{Detected the `minimal' class.\MessageBreak
1048     Expect lots of warnings and some malfunctions.\MessageBreak
1049     You might want to use a proper class instead}%
1050 }\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`.

```

1051 </package>
1052 <*package|letterspace>
1053 <plain>\MT@requires@latexl{
1054 \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.
1055 \def\MT@addto@setup{\g@addto@macro\MT@setup@}
                  Don't hesitate with miniltx.
1056 <plain>{\let\MT@addto@setup\@firstofone}

\MT@with@package@T    We almost never do anything if a package is not loaded.
1057 \def\MT@with@package@T#1{\ifpackageloaded{#1}\@firstofone\@gobble}
1058 </package|letterspace>
1059 <*package>

\MT@with@babel@and@T    LATEX's \ifpackagewith ignores the class options.
1060 \def\MT@with@babel@and@T#1{%
1061   \MT@ifdefined@n@T{opt@babel.sty}{%
1062     \expandtwoargs\MT@inclist{#1}
1063     {\csname opt@babel.sty\endcsname,\@classoptionslist}%
1064     \ifMT@inlist@expandafter\@secondoftwo\else\expandafter\@firstofone\fi
1065   }\@gobble
1066 }

\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 XYTEX. The successor
                   packages eledmac and reledmac are also supported.
1067 </package>
1068 <pdf->\MT@requires@pdftex5{
1069 <*pdf-|lua-|xe->
1070   \def\MT@ledmac@setup{%
1071     \ifMT@protrusion
1072       \MT@ifdefined@c@TF\l@dunhbox@line{%
\MT@led@unhbox@line    Hook.
1073         \MT@info@n1{Patching ((r)e)ledmac to enable character protrusion}%
1074         \let\MT@led@unhbox@line\l@dunhbox@line
1075         \renewcommand*\l@dunhbox@line}[1]{%
1076           \ifhbox##1%
1077             \kern\leftmarginkern##1%
1078             \expandafter\MT@led@unhbox@line\expandafter##1\expandafter
1079             \kern\rightmarginkern##1%
1080           \fi
1081         }%
1082       }{%
1083         \MT@warning@n1{%
1084           Character protrusion in paragraphs with line\MessageBreak
1085           numbering will only work if you update ledmac,\MessageBreak
1086           or use one of its successors, eledmac or reledmac}%
1087       }%
1088     \fi
1089   }
1090 </pdf-|lua-|xe->
1091 <*pdf->
1092 }{
1093   \def\MT@ledmac@setup{%
1094     \ifMT@protrusion

```

```

1095     \MT@warning@n1{%
1096         The pdftex version you are using does not allow\MessageBreak
1097         character protrusion in paragraphs with line\MessageBreak
1098         numbering by the ~((r)e)ledmac' package.\MessageBreak
1099         Upgrade pdftex to version 1.30 or later}%
1100     \fi
1101 }
1102 }
1103 /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 `\#`.

```

1104 (*package|letterspace)
1105 (*package)
1106 \def\MT@restore@p@h{\chardef\%~\% \chardef\#~\# }

```

`\ifMT@fontspec` Two new conditionals for use with X_YTeX or LuaTeX.

```

\ifMT@xunicode 1107 \newif\ifMT@fontspec
1108 \MT@with@package@T{fontspec}\MT@fontspectrue
1109 \newif\ifMT@xunicode
1110 \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).

```

1111 \IfFormatAtLeastTF{2020/10/01}
1112   {\IfFormatAtLeastTF{2021/11/15}
1113     {\AddToHook{package/fontspec/after}{\MT@fontspectrue}}
1114     {\AddToHook{package/after/fontspec}{\MT@fontspectrue}}\relax

```

`\MT@maybe@gobble@with@tikz` 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`.

`\MT@tikz@setup`

```

1115 \let\MT@maybe@gobble@with@tikz\@firstofone
1116 \def\MT@tikz@setup{%
1117   \def\MT@maybe@gobble@with@tikz{%
1118     \ifnum\tikz@expandcount>\z@
1119       \expandafter\@gobble
1120     \else
1121       \expandafter\@firstofone
1122     \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`.)

```

1123 \def\MT@setupfont@hook{%

```

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

```

1124 \MT@if@false
1125 \MT@with@babel@and@T{spanish} \MT@if@true
1126 \MT@with@babel@and@T{galician}\MT@if@true
1127 \MT@with@babel@and@T{mexican} \MT@if@true
1128 \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.)

```
1129 \MT@with@package@T{csquotes}{%
1130 \ifpackage@later{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`.

```
1131 \MT@if@false
1132 \MT@with@package@T{hyperref} \MT@if@true
1133 \MT@with@package@T{tex4ht} \MT@if@true
1134 \MT@with@package@T{mathastext}\MT@if@true
1135 \ifMT@if@MT@restore@p@h\fi
1136 \MT@with@package@T{tikz}\MT@tikz@setup
1137 }
```

Check again at the end of the preamble.

```
1138 </package>
1139 \MT@addto@setup{%
1140 *package>
```

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

```
1141 \MT@with@package@T{pdfcpot}{%
1142 \MT@error{Detected the `pdfcpot' package!\MessageBreak
1143 \MT@MT' and `pdfcpot' may not be used together}{%
1144 The `pdfcpot' package provides an interface to character protrusion.\MessageBreak
1145 So does the `MT@MT' package. Using both packages at the same\MessageBreak
1146 time will almost certainly lead to undesired results. Have your choice!}%
1147 }%
1148 \MT@with@package@T {ledmac}\MT@ledmac@setup
1149 \MT@with@package@T {eledmac}\MT@ledmac@setup
1150 \MT@with@package@T{reledmac}\MT@ledmac@setup
1151 \MT@with@package@T{xunicode}\MT@xunicodetrue
1152 \MT@with@package@T{fontspec}\MT@fontspec@true
```

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

```
1153 \MT@gl@et\MT@setupfont@hook\@empty
```

`microtype` is so so loquacious ... Sometimes you just want to silence it when debugging a document.⁴

```
1154 %\gdef\MT@setupfont@hook{\ifnum\tracingmacros>\z@\tracingnone
1155 % \MT@info{-->Silently doing my `magic' (Mittelbach) for font\MessageBreak->\MT@font}\fi}%
1156 \MT@if@false
1157 \MT@with@babel@and@T{spanish} \MT@if@true
1158 \MT@with@babel@and@T{galician}\MT@if@true
1159 \MT@with@babel@and@T{mexican} \MT@if@true
1160 \ifMT@if@
1161 \g@addto@macro\MT@setupfont@hook{%
1162 \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
1163 \fi
1164 \MT@with@package@T{csquotes}{%
1165 \ifpackage@later{csquotes}{2005/05/11}{%
1166 \g@addto@macro\MT@setupfont@hook\@disablequotes
```

For `\leftprotrusion`, we disable `csquotes`'s tracking of group level and type, because we'll probably be typesetting the opening quotes only.

```
1167 \g@addto@macro\MT@prot@hook{%
1168 \def\csq@bqgroup{\begingroup\leavevmode
1169 \let\MT@csq@eqgroup\endgroup}%
1170 \let\csq@eqgroup\endgroup}%
1171 }{%
1172 \MT@warning@n1{%
```

4 Cf. <https://www.youtube.com/watch?v=7FQLnggVgDE&t=38m24s>

```

1173     Should you receive warnings about unknown slot\MessageBreak
1174     numbers, try upgrading the `csquotes' package}%
1175   }%
1176 }%

```

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.

```

1177 \MT@if@false
1178 </package>
1179 <plain> \MT@requires@latex2{
1180   \MT@with@package@T{hyperref}{%
1181     \pdfstringdefDisableCommands{%
1182 <*package>
1183       \MT@ltx@pickupfont
1184       \let\textmicrotypecontext\@secondoftwo
1185       \let\microtypecontext\@gobble
1186 </package>
1187       \def\lssstyle{\pdfstringdefWarn\lssstyle}%
1188       \def\textls#1#\pdfstringdefWarn\textls}%
1189     }%
1190 <package> \MT@if@true
1191   }%
1192 <plain> } \relax
1193 <*package>
1194   \MT@with@package@T{tex4ht}{%
1195     \def\MT@apply@patch#1{\MT@info{Not applying patch `#1' (for tex4ht)}}%
1196     \def\MT@undo@patch#1{\MT@info{Not undoing patch `#1' (for tex4ht)}}%
1197     \MT@if@true
1198   }%
1199   \MT@with@package@T{mathastext}\MT@if@true
1200   \ifMT@if@vg@addto@macro\MT@setupfont@hook\MT@restore@p@h@fi

```

The `listings` package makes numbers and letters active,

```

1201 \MT@with@package@T{listings}{%
1202   \g@addto@macro\MT@cfg@catcodes{%
1203     \MT@while@num{"30"}{"3A"}{\catcode\@tempcnta=12\relax}%
1204     \MT@while@num{"41"}{"5B"}{\catcode\@tempcnta=11\relax}%
1205     \MT@while@num{"61"}{"7B"}{\catcode\@tempcnta=11\relax}%
1206   }%

```

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

```

1207   \g@addto@macro\MT@setupfont@hook{%
1208     \catcode`\=\z@

```

Inside a listing, `\space` is redefined.

```

1209     \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.

```

1210     \let\lst@ProcessLetter\@empty
1211   }%
1212 }%

```

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.

```

1213 </package>
1214 \ifx\SOULE@\undefined\else
1215   \soulregister\lssstyle 0%

```

```

1216 \soulregister\textls 1%
1217 \ifx\XeTeXrevision\undefined
1218 \let\MT@S@S@O@L@doword\SOUL@doword
1219 \def\SOUL@doword{\pdfadjustspacing=\z@ \MT@S@O@L@doword}%
1220 \fi
1221 \fi
1222 *package
1223 \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.

```

1224 \MT@with@package@T{pinyin}{%
1225 \let\MT@orig@py@macron\py@macron
1226 \@ifpackage@later{pinyin}{2005/08/11}{% 4.6.0
1227 \def\py@macron#1#2{%
1228 \MT@!tx@pickupfont
1229 \MT@orig@py@macron{#1}{#2}%
1230 \MT@MT@pickupfont}%
1231 }{%
1232 \def\py@macron#1{%
1233 \MT@!tx@pickupfont
1234 \MT@orig@py@macron{#1}%
1235 \MT@MT@pickupfont}%
1236 }%
1237 }%

```

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`.

```

1238 \MT@with@package@T{unicode-math}{%
1239 \MT@!et@enc{__um_sub_or_super:n}\relax
1240 }%
1241 /package
1242 }
1243 *package

```

1.1.7 Protrusion patches

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

```

\MT@patch@info 1244 \newif\ifMT@patch@ok
\MT@patch@warn 1245 \def\MT@patch@info#1{\MT@info{Applying patch `#1'}}
1246 \def\MT@patch@warn#1{\MT@warning{Unable to apply patch `#1'}}
\MT@patch@undef 1247 \def\MT@patch@undef#1{\MT@warning{Patch `#1' undefined.\MessageBreak Cannot apply it}}
\MT@patch@info@undo 1248 \def\MT@patch@info@undo#1{\MT@info{Reverting patch `#1'}}

```

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

```

\MT@define@patch
1249 \let\MT@patches@def\@gobble
1250 \def\MT@define@patch#1#2#3{%
1251 \MT@ifdefined@n@TF{MT@patch@#1}{%
1252 \MT@warning{Patch `#1' already defined.\MessageBreak Cannot define it}%
1253 }{%
1254 \g@addto@macro\MT@patches@def{,#1}%
1255 \MT@gdef@n{MT@patch@#1}{#2}%
1256 \MT@gdef@n{MT@patch@undo@#1}{#3}%
1257 }%
1258 }

```


`\MT@redefined@patches` 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).

`\MT@redefine@patch`

```
1259 \let\MT@redefined@patches\@empty
1260 \def\MT@redefine@patch#1#2#3{%
1261   \g@addto@macro\MT@redefined@patches{%
1262     \MT@ifdefined@n@TF{MT@patch@#1}{%
1263       \MT@gdef@n{MT@patch@#1}{#2}%
1264       \MT@gdef@n{MT@patch@undo@#1}{#3}%
1265     }%
1266     \MT@warning{Patch `#1' undefined.\MessageBreak Cannot redefine it}%
1267   }%
1268 }%
1269 }
```

Both macros are only allowed in the preamble.

```
1270 \@onlypreamble\MT@define@patch
1271 \@onlypreamble\MT@redefine@patch
```

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

`\MT@patch@patch`

```
1272 \def\MT@append@patch#1#2{%
1273   \MT@remember@patch{#1}%
1274   \apptocmd#1{#2}\relax\MT@patch@okfalse
1275 }
1276 \def\MT@patch@patch#1#2#3{%
1277   \MT@remember@patch{#1}%
1278   \patchcmd#1{#2}{#3}\relax\MT@patch@okfalse
1279 }
```

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

```
1280 \def\MT@remember@patch#1{%
1281   \MT@ifdefined@n@TF{MT@patch@saves@string#1}\relax
1282   {\MT@let@nc{MT@patch@saves@string#1}#1%
1283     \MT@exp@cs\g@addto@macro{MT@patch@undo@@\MT@patch@name}%
1284     {\MT@let@cn#1{MT@patch@saves@string#1}}%
1285 }
```

`\MT@patches@applied` 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).

`\MT@apply@patch`

```
1286 \let\MT@patches@applied\@gobble
1287 \def\MT@apply@patch#1{%
1288   \MT@patch@oktrue
1289   \MT@ifdefined@n@TF{MT@patch@#1}
1290   {\MT@in@clist{#1}\MT@patches@applied
1291     \ifMT@in@list@
1292       \MT@warning{Patch `#1' has already been applied,\MessageBreak
1293         cannot reapply it}%
1294     \else
1295       \let\MT@restore@catcodes\@empty
1296       \MT@with@babel@and@T{spanish} {\MT@fix@catcode{62}{12}}% >
1297       \MT@with@babel@and@T{galician} {\MT@fix@catcode{62}{12}}% >
1298       \def\MT@patch@name{#1}%
1299       \g@addto@macro\MT@patches@applied{,#1}%
1300       \@nameuse{MT@patch@#1}%
1301       \@nameuse{MT@patch@\ifMT@patch@ok info\else warn\fi}{#1}%
1302       \MT@restore@catcodes
1303       \fi
1304     {\MT@patch@undef{#1}}%
1305 }
```

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

```
1306 \def\MT@undo@patch#1{%
1307   \MT@in@clist{#1}\MT@patches@applied
1308   \ifMT@in@list@
```

```

1309 \MT@rem@from@clist{#1}\MT@patches@applied
1310 \@nameuse{MT@patch@undo@#1}%
1311 \MT@patch@info@undo{#1}%
1312 \else
1313 \MT@warning{Patch `#1' hasn't been applied,\MessageBreak cannot revert it}%
1314 \fi
1315 }

```

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.

```

1316 {\catcode`\#=12
1317 \MT@addto@setup{%

```

Now for the actual patches:

`\@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

```

1318 \MT@define@patch{item}{%
1319 \MT@append@patch{\@item\leftprotrusion
1320 \MT@patch@patch{\@item\everypar{}}{\everypar{\leftprotrusion}}%

```

- beamer patches it too

```

1321 \ifclassloaded{beamer}
1322 {\MT@append@patch\beamer@callorigitem\leftprotrusion
1323 \MT@patch@patch\beamer@callorigitem\ignorespaces\leftprotrusion}}

```

- the `simplecv` class

```

1324 \ifclassloaded{simplecv}
1325 {\MT@append@patch\@topic@item\leftprotrusion}
1326 {}}%
1327 }{}%

```

`toc`: TOC and friends

```

1328 \MT@define@patch{toc}{%
1329 \MT@append@patch\numberline\leftprotrusion

```

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

```

1330 \ifclassloaded{memoir}
1331 {\MT@append@patch\booknumberline\leftprotrusion
1332 \MT@append@patch\partnumberline\leftprotrusion
1333 \MT@append@patch\chapternumberline\leftprotrusion
1334 \MT@append@patch\cftbookafterpnum\noprotrusion
1335 \MT@append@patch\cftpartafterpnum\noprotrusion
1336 \MT@append@patch\cftchapterafterpnum\noprotrusion
1337 \MT@append@patch\cftsectionafterpnum\noprotrusion
1338 \MT@append@patch\cftsubsectionafterpnum\noprotrusion
1339 \MT@append@patch\cftsubsubsectionafterpnum\noprotrusion
1340 \MT@append@patch\cftparagraphafterpnum\noprotrusion
1341 \MT@append@patch\cftsubparagraphafterpnum\noprotrusion
1342 \MT@append@patch\cftfigureafterpnum\noprotrusion
1343 \MT@append@patch\cfttableafterpnum\noprotrusion}
1344 {}}%
1345 }{}%

```

- 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 `\noprotusion`. 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.

```

1346 % \ifpackageloaded{tocbasic}
1347 % {\MT@define@patch{toc}
1348 % {\MT@append@patch\numberline\leftprotrusion
1349 % \setuptoc{toc}{noprotrusion}%
1350 % \setuptoc{lof}{noprotrusion}%
1351 % \setuptoc{lot}{noprotrusion}}
1352 % {\unsettoc{toc}{noprotrusion}%
1353 % \unsettoc{lof}{noprotrusion}%
1354 % \unsettoc{lot}{noprotrusion}}}%

```

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

eqnum: equation numbers

- IEEEtran

```

1355 \MT@define@patch{eqnum}{%
1356 \ifclassloaded{IEEEtran}
1357 {\MT@patch@patch\theequationdis{()}\leftprotrusion{()}}%
1358 \MT@patch@patch\theequationdis{()}\rightprotrusion{()}}%
1359 \MT@patch@patch\theIEEEsubequationdis{()}\leftprotrusion{()}}%
1360 \MT@patch@patch\theIEEEsubequationdis{()}\rightprotrusion{()}}%
1361 {}%

```

- `\eqref` (amsmath) relies on `\tagform@`, so we have to have it use the original definition.

```

1362 \ifpackageloaded{amsmath}
1363 {\MT@patch@patch\tagform@{()}\leftprotrusion{()}}%
1364 \MT@patch@patch\tagform@{()}\rightprotrusion{()}}%

```

The command has been made robust in 2022.

```

1365 \MT@ifdefined@n@TF{eqref }
1366 {\MT@exp@cs\MT@patch@patch{eqref }}{\MT@patch@patch\eqref }
1367 {\tagform@}{\@nameuse{MT@patch@saved@string\tagform@}}%

```

- If the user has altered the tags' appearance via `mathtools`'s `\newtagform` interface, our patch won't have any effect. We don't issue a warning because `\(left|right)protrusion` might have been specified appropriately in `\newtagform`. We could also patch the latter command (or, to be more precise, `\MT_define_tagform:nwnn`), but the timing is a bit tricky, so for now info it is.

```

1368 \MT@with@package@T{mathtools}{%
1369 \ifMT@patch@ok\else \MT@patch@oktrue
1370 \MT@info@nl{The `eqnum' patch may not be effective because you are\MessageBreak
1371 using the mathtools package. Make sure to insert\MessageBreak
1372 `\

```

footnote: footnote text (only visible with block paragraphs)

- `hyperref` also patches this command (but only if `hyperfootnotes=true`, `implicit=true` and `\hyper@nopatch@footnote` is undefined)

```

1379 \MT@define@patch{footnote}{%

```

```

1380 \ifpackageloaded{hyperref}
1381   {\MT@if@false
1382     \ifHy@implicit
1383     \ifHy@hyperfootnotes
1384     \MT@ifdefined@c@TF\hyper@nopatch@footnote\relax
1385     \MT@if@true
1386     \fi
1387     \fi
1388     \ifMT@if@\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
1389   \@secondoftwo
1390   {\MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
1391     \MT@patch@patch\@footnotetext{\@empty\ignorespaces}{\@empty\ignorespaces\leftprotrusion}%
1392     \MT@patch@patch\@mpfootnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
1393     \MT@patch@patch\@mpfootnotetext
1394       {\expandafter\hyper@@anchor\expandafter
1395         {\Hy@footnote@currentHref}{\relax}}\ignorespaces}
1396     {\expandafter\hyper@@anchor\expandafter
1397       {\Hy@footnote@currentHref}{\relax}}\ignorespaces\leftprotrusion}}

```

- memoir additionally allows footnotes in the margins

```

1398   {\@ifclassloaded{memoir}
1399     {\MT@patch@patch\@footnotetext{\foottextfont #1}{\foottextfont\leftprotrusion #1}%
1400     \MT@patch@patch\@mpfootnotetext{\foottextfont #1}{\foottextfont\leftprotrusion #1}}

```

- beamer has it its own way, of course

```

1401   {\@ifclassloaded{beamer}
1402     {\MT@exp@cs\MT@patch@patch{beamerx@\string\beamer@framefootnotetext}
1403       {\ignorespaces}{\ignorespaces\leftprotrusion}%
1404     \MT@exp@cs\MT@patch@patch{beamerx@\string\@mpfootnotetext}
1405       {\ignorespaces}{\ignorespaces\leftprotrusion}}

```

- the KOMA classes

```

1406     {\MT@ifdefined@c@TF\KOMAClassName
1407     {\MT@patch@patch\scr@saved@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}%

```

- the base classes

```

1408     {\MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}%
1409     \MT@patch@patch\@mpfootnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}}%
1410   }{}%

```

verbatim: disable all microtypographic extensions in verbatim blocks. (This could have been a nice opportunity to use the new L^AT_EX hook management, however, the hook here is executed too early – namely, before the `\par` in `\@verbatim`, which may result in spilling the microtypographic settings to the preceding paragraph – so we’re resorting to patching, again.)

- Appending to `\@verbatim` works for, at least, the standard classes, `verbatim` (and `memoir`); the implementations in `fancvrb` and `listings` don’t allow protrusion anyway.

```

1411   \MT@define@patch{verbatim}{%
1412     \MT@append@patch\@verbatim{\microtypesetup{activate=false}}%

```

- package `alltt`

```

1413     \MT@with@package@T{alltt}{\MT@append@patch\alltt{\microtypesetup{activate=false}}}%
1414   }{}

```

Finally, execute any redefinitions.

```

1415   \MT@redefined@patches
1416 }{}
1417 </package>
1418 </package|letterspace>

```

1.2 Font setup

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

```
1419 <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`).

```
1420 <*pdf-|lua-|xe->
```

```
1421 \def\MT@setupfont{%
```

With X_YTeX 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.

```
1422 <xe-|lua-> \MT@font
```

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

```
1423 \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).

```
1424 <pdf->\MT@requires@pdftex7{
```

```
1425 <pdf-|lua->\g@addto@macro\MT@setupfont\MT@copy@font
```

```
1426 <pdf->}\relax
```

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

```
1427 \g@addto@macro\MT@setupfont{%
```

```
1428 \MT@exp@two@c\MT@split@name\string\MT@font/\@nil
```

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

```
1429 \MT@exp@one@n\MT@find@file\MT@family
```

```
1430 \ifx\MT@familyalias\@empty \else
```

```
1431 \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.)

```
1432 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
```

```
1433 }
```

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

```
1434 <pdf->\MT@requires@pdftex6
```

```
1435 <lua->\MT@requires@luatex3
```

```
1436 <pdf-|lua-> {\g@addto@macro\MT@setupfont\MT@tracking}\relax
```

```
1437 \g@addto@macro\MT@setupfont{%
```

```
1438 \MT@check@font
```

```
1439 \ifMT@inlist@
```

```
1440 <debug>\MT@show@pdfannot2%
```

```
1441 \else
```

```
1442 \MT@vinfo{Setting up font `~\MT@@font'\on@line}%
```

```
1443 \MT@info@nottracking
```

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.

```
1444 \MT@protrusion
```

```
1445 <pdf-|lua-> \MT@expansion
```

```
1446 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
1447 <pdf-
1448 \MT@requires@pdftex6{
1449 \g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}
1450 }\relax
1451 </pdf-
```

Disable ligatures (pdfTeX 1.30).

```
1452 <pdf-\MT@requires@pdftex5{
1453 <pdf-|lua-\g@addto@macro\MT@setupfont\MT@noligatures
1454 <pdf-}\relax
1455 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
1456 <debug\MT@show@pdfannot1%
```

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

```
1457 \MT@register@font
1458 \fi
1459 }
1460 </pdf-|lua-|xe-
```

`\MT@copy@font` The new (1.40.4) `\pdfcopyfont` command allows expanding a font with different 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.

```
1461 <pdf-|lua-
1462 \let\MT@copy@font\relax
1463 <pdf-\MT@requires@pdftex7{
1464 \def\MT@copy@font{%
```

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

```
1465 \xdef\MT@font@copy{\csname\MT@font/\MT@pr@context/\MT@ex@context\endcsname}%
1466 \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.

```
1467 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
1468 \expandafter\ifx\MT@font@orig\relax
1469 \MT@exp@two@c\MT@gl@t\MT@font@orig\font@name
1470 \else
1471 \MT@exp@two@c\let\font@name\MT@font@orig
1472 \fi
1473 <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.

```
1474 <lua-\MT@exp@two@c\MT@lua@copyfont\meaning\font@name\@nil
1475 <debug\MT@dinfo1{creating new copy: \MT@font@copy}%
```

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

```
1476 \MT@map@clist@c\MT@active@features{
1477 \MT@exp@cs@ifx\MT@\@nameuse\{MT@abbr@##1}\relax\else
1478 \def\@tempa{##1}%
1479 \MT@exp@cs\MT@map@tlist@c\{MT@##1@doc@contexts}\MT@rem@from@list
1480 \fi
1481 }%
1482 \fi
1483 \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

```
1484 \let\font@name\MT@font@copy
```

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

```
1485 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
1486 }
```

```
\MT@rem@from@list
```

```
1487 \def\MT@rem@from@list#1{%
1488   \MT@exp@cs\ifx\MT@\@tempa @#1font@list}\relax\else
1489   \expandafter\MT@exp@one@n\expandafter\MT@rem@from@list\expandafter
1490   \MT@font \csname MT@\@tempa @#1font@list\endcsname
1491   \fi
1492 }
1493 \pdf-}\relax
```

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

```
1494 \lua-\def\MT@lua@copyfont #1 #2 #3\nil{%
1495 \lua-\global\expandafter\font\MT@font@copy=#3\relax}
1496 \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 parameters 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
\MT@dimen@six
```

If `\fontdimen 6` is zero, character protrusion, spacing, kerning and tracking won't 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 (and since pdfTeX 1.40.23, this also works for `\letterspacefont`). XeTeX (and newer LuaTeX in DVI mode) doesn't provide an equivalent to `\pdffontsize`, so we use the nominal size instead.

```
1497 \pdf-|\lua-|\xe-
1498 \def\MT@fix@fontdimen@six{%
1499   \ifnum\fontdimen6\MT@font=\z@
1500     \fontdimen6\MT@font=%
1501     \pdffontsize\MT@font
1502   \lua-\MT@requires\luatex4{\ifnum\outputmode=\@ne \pdffeedback fontsize\else
1503   \lua-|\xe-           \MT@size pt%
```

```

1504 <lua-> \expandafter\@gobble\fi}{\pdffontsize}\MT@font
1505 \MT@info{Fixing zero \@backslashchar fontdimen 6 for font \MT@font'\MessageBreak
1506 (new value: \the\fontdimen6\MT@font)}%
1507 <pdf-> \MT@requirespdfTeX8\relax{\MT@gletenc{\MT@font-fake6}\@empty}%
1508 \fi
1509 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1510 }
1511 </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 1512 <*package>
\MT@shape 1513 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
1514 \def\MT@encoding{#1}%
\MT@size 1515 \ifMT@fontspec
1516 \edef\MT@family{\MT@scrubfeature#2()\relax}%
1517 \else
1518 \def\MT@family{#2}%
1519 \fi
1520 \def\MT@series {#3}%
1521 \def\MT@shape {#4}%
1522 \def\MT@size {#5}%
1523 \MT@fix@fontdimen@six
\MT@familyalias Alias family?
1524 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
1525 {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
1526 {\let\MT@familyalias\@empty}%
1527 }
\MT@scrubfeature Remove one resp. all feature counters (fontspec).
\MT@scrubfeatures 1528 \def\MT@scrubfeature#1(#2)#3\relax{#1}
1529 \def\MT@scrubfeatures#1(#2)#3\relax{%
1530 #1%
1531 \ifx\relax#3\relax\else
1532 \MT@scrubfeatures#3\relax
1533 \fi
1534 }
\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 1535 \newif\ifMT@do
1536 \def\MT@maybe@do#1{%
(but only if the feature isn't globally set to false)
1537 \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).
1538 \MT@dotrue
1539 \edef\@tempa{\csname MT@#1@setname\endcsname}%
1540 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1541 \MT@ifdefined@n@TF{MT@checklist@#1}%
1542 {\csname MT@checklist@#1\endcsname}%
1543 {\MT@checklist@{#1}}}%
1544 {#1}%
1545 }%
1546 \else
1547 \MT@dofalse
1548 \fi
1549 \ifMT@do

```


\MT@feat stores the current feature.

```

1550 \def\MT@feat{#1}%
1551 \csname MT@set@#1@codes\endcsname
1552 \else
1553 \MT@ifstreq{#1}{tr}%
1554 {\let\MT@info@nottracking\MT@info@nottracking@}%
1555 {\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}%
1556 \fi
1557 }

```

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

```

\MT@info@nottracking@ 1558 \let\MT@info@nottracking\relax
1559 \def\MT@info@nottracking@{\MT@vinfo{... No tracking}}

```

\MT@dinfo@list

```

1560 <debug>\def\MT@dinfo@list#1#2#3{\MT@dinfo@n1}{\@nameuse{MT@abbr@#1}: #2
1561 <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).

```

1562 \def\MT@checklist@#1#2{%
1563 <!debug> \MT@ifdefined@n@T
1564 <debug> \MT@ifdefined@n@TF
1565 {\MT@#2list@#1@\@tempa}{%

```

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

```

1566 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
1567 \csname MT@#1\expandafter\endcsname
1568 \csname MT@#2list@#1@\@tempa\endcsname
1569 \ifMT@inlist@
1570 <debug>\MT@dinfo@list{#2}{#1}{in}%
1571 \MT@dotrue
1572 \else
1573 <debug>\MT@dinfo@list{#2}{#1}{not in}%
1574 \MT@dofalse
1575 \expandafter\MT@clist@break
1576 \fi
1577 }%

```

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.

```

1578 <debug> {\MT@dinfo@list{#2}{#1}{}}%
1579 }

```

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

```

1580 \def\MT@checklist@family#1{%
1581 <!debug> \MT@ifdefined@n@T
1582 <debug> \MT@ifdefined@n@TF
1583 {\MT@#1list@family@\@tempa}{%
1584 \MT@exp@two@n\MT@in@clist
1585 \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
1586 \ifMT@inlist@
1587 <debug>\MT@dinfo@list{#1}{family}{in}%
1588 \MT@dotrue
1589 \else
1590 <debug>\MT@dinfo@list{#1}{family}{not in}%
1591 \MT@dofalse
1592 \ifx\MT@familyalias\empty \else
1593 \MT@exp@two@n\MT@in@clist
1594 \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
1595 \ifMT@inlist@
1596 <debug> \MT@dinfo@list{#1}{family alias}{in}%
1597 \MT@dotrue
1598 <debug>\else\MT@dinfo@list{#1}{family alias}{not in}%

```

```

1599     \fi
1600     \fi
1601     \fi
1602     \ifMT@do \else
1603     \expandafter\MT@clist@break
1604     \fi
1605     }%
1606     <debug> {\MT@dinfo@list{#1}{family}{}}%
1607 }

```

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

```

1608 \def\MT@checklist@size#1{%
1609 <!debug> \MT@ifdefined@n@T
1610 <debug> \MT@ifdefined@n@TF
1611     {MT@#1list@size@@tempa}%
1612     \MT@exp@cs\MT@in@rlist{MT@#1list@size@@tempa}%
1613     \ifMT@inlist@
1614 <debug>\MT@dinfo@list{#1}{size}{in}%
1615     \MT@dotrue
1616     \else
1617 <debug>\MT@dinfo@list{#1}{size}{not in}%
1618     \MT@dofalse
1619     \expandafter\MT@clist@break
1620     \fi
1621     }%
1622 <debug> {\MT@dinfo@list{#1}{size}{}}%
1623 }

```

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

```

1624 \def\MT@checklist@font#1{%
1625 <!debug> \MT@ifdefined@n@T
1626 <debug> \MT@ifdefined@n@TF
1627     {MT@#1list@font@@tempa}%

```

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

```

1628     \edef@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1629     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1630     \@tempb \csname MT@#1list@font@@tempa\endcsname
1631     \ifMT@inlist@
1632 <debug>\MT@dinfo@list{#1}{font}{in}%
1633     \expandafter\MT@clist@break
1634     \else
1635 <debug>\MT@dinfo@list{#1}{font}{not in}%
1636     \MT@dofalse
1637     \fi
1638     }%
1639 <debug> {\MT@dinfo@list{#1}{font}{}}%
1640 }

```

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`.

```
1641 \newif\ifMT@nofamily
```

`\MT@protrusion` Set up for protrusion?

```
1642 \def\MT@protrusion{\MT@maybe@do{pr}}
1643 </package>
```

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

```
1644 <*pdf-|lua-|xe-|show>
```

```

1645 <show>\def\MTS@show@pr
1646 <pdf-|lua-|xe->\def\MT@set@pr@codes
1647 {%
1648 <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).

```

1649 <show> \MTS@printtext{Protrusion settings for font '\texttt{\MT@font}':}\
1650 \MT@if@list@exists{%
1651 <*pdf-|lua-|xe->
1652 \ifMT@nofamily
1653 \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
1654 \MT@info@n@l{Loading generic protrusion settings for font family\MessageBreak
1655 \MT@family' (encoding: \MT@encoding).\MessageBreak
1656 For optimal results, create family-specific settings.\MessageBreak
1657 See the microtype manual for details}%
1658 \MT@glet@nc{\MT@encoding-\MT@family-settings}\@empty
1659 }%
1660 \fi
1661 </pdf-|lua-|xe->
1662 <show> \MTS@printtext{First matching list is for '\texttt{\@tempa}':\
\texttt{\MT@pr@c@name}}%
1663 \MT@get@opt
1664 \MT@reset@pr@codes

```

Get the name of the inheritance list and parse it.

```
1665 \MT@get@inh@list
```

Set an input encoding?

```
1666 \MT@set@inputenc{c}%
```

Load additional lists?

```
1667 \MT@load@list\MT@pr@c@name
1668 \MT@set@listname

```

Load the main list.

```

1669 \MT@let@cn\@tempc{\MT@pr@c@\MT@pr@c@name}%
1670 \expandafter\MT@set@codes\@tempc,\relax,%
1671 <show> \vrule width 4cm height .5pt \
1672 <show> \MTS@printtext{End of list '\texttt{\MT@pr@c@name}'}\
\ [.5em]
1673 <show> \MT@ifdefined@c@T\MT@pr@inh@name{%
1674 <show> \MT@ifdefined@n@T{\MT@inh@\MT@pr@inh@name @prefixes}{%
1675 <show> \par \MTS@printtext{(with prefixes:)}%
1676 <show> \@tempcntb=\z@

```

Set unconditional heirs.

```

1677 \MT@set@pr@prefixheirs
1678 <show> }%
1679 <show> \ifShowMissingGlyphs\MTS@show@missing\fi
1680 }%
1681 <show> {\MTS@printtext{NOT DEFINED}}%
1682 \MT@reset@pr@codes
1683 <show> }\par
1684 }

```

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

```

1685 <*pdf-|lua-|xe->
1686 \def\MT@set@all@pr#1#2{%
1687 <debug>\MT@dinfol{3}{-- lp/rp: setting all to #1/#2}%
1688 \let\MT@temp\@empty
1689 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1}}%
1690 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2}}%
1691 \MT@do@font\MT@temp
1692 }

```

\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.

```
1693 \def\MT@reset@pr@codes@{\MT@set@all@pr@z@z@}
1694 \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`.

```
1695 \def\MT@the@pr@code{\@tempcntb}
1696 <pdf-|lua-
1697 <pdf-)\MT@requires@pdftex6
1698 <lua-)\MT@requires@luatex3
1699 {\def\MT@the@pr@code@tr{%
1700 \numexpr\@tempcntb+\MT@letterspace@/2\relax
1701 }
1702 }\relax
1703 </pdf-|lua-
```

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

```
1704 \def\MT@set@codes#1,{%
1705 \ifx\relax#1\@empty\else
1706 \MT@split@codes #1==\relax
1707 \expandafter\MT@set@codes
1708 \fi
1709 }
```

`\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.

```
1710 \def\MT@split@codes#1=#2=#3\relax{%
1711 \def\@tempa{#1}%
1712 \ifx\@tempa\@empty \else
1713 \MT@get@slot
1714 <pdf-|lua- \ifnum\MT@char > \m@ne
1715 <xe- \ifx\MT@char\@empty \else
1716 \MT@get@char@unit
1717 \csname MT@\MT@feat @split@val\endcsname#2\relax
1718 \fi
1719 \fi
1720 }
```

`\MT@pr@split@val`

```
1721 \def\MT@pr@split@val#1,#2\relax
1722 </pdf-|lua-|xe-
1723 <show)\def\MTS@pr@split@val#1,#2\relax
1724 {\def\@tempb{#1}%
1725 \MT@ifempty\@tempb
1726 <pdf-|lua-|xe- \relax
1727 <show) {\MTS@lp@=\z@ \let\MTS@lpcode\@empty}%
1728 {\MT@scale@to@em
1729 <pdf-|lua-|xe- \lpcode\MT@font\MT@char=\MT@the@pr@code
1730 <show) \MTS@lp@=\dimexpr\@tempcntb em/1000\relax\relax
1731 <show) \edef\MTS@lpcode{[\@tempb] \the\@tempcntb/\the\MTS@lp@}%
1732 <debug)\MT@dinfo@n1{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
1733 }%
1734 \def\@tempb{#2}%
1735 \MT@ifempty\@tempb
1736 <pdf-|lua-|xe- \relax
1737 <show) {\MTS@rp@=\z@ \let\MTS@rpcode\@empty}%
1738 {\MT@scale@to@em
1739 <pdf-|lua-|xe- \rancode\MT@font\MT@char=\MT@the@pr@code
1740 <show) \MTS@rp@=\dimexpr\@tempcntb em/1000\relax\relax
1741 <show) \edef\MTS@rpcode{[\@tempb] \the\@tempcntb/\the\MTS@rp@}%
1742 <debug)\MT@dinfo@n1{4}{;;; rp (\MT@char): \number\rancode\MT@font\MT@char: [#2]}%
```

```

1743 }%
1744 <show> \llap{\MT@show@char@pr\MT@char\quad}%
1745 <show> \parbox[b]{}[b]{3.5cm}{\MT@printtext}%
1746 <show> \footnotesize\makebox[.4cm][l]{L:} \MT@ifempty{\MT@lpcode}{---}{\MT@lpcode}\
1747 <show> \makebox[.4cm][r]{R:} \MT@ifempty{\MT@rpcodes}{---}{\MT@rpcodes}}%
1748 <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>@`.

```

1749 \MT@ifdefined@cT\MT@pr@inh@name{%
1750 \MT@ifdefined@nT\MT@inh@\MT@pr@inh@name @\MT@char @}%
1751 \MT@exp@cs\MT@map@tlist@c
1752 {MT@inh@\MT@pr@inh@name @\MT@char @}%
1753 <pdf-|lua-|xe- > \MT@set@pr@heirs
1754 <show> \MT@show@char@pr
1755 }%
1756 }%
1757 <show> }\newline
1758 }
1759 <*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. `\rpcodes`, since this would disallow protrusion factors larger than the character width (since `\lrcode`’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`.

```

1760 <pdf- > \MT@requires@pdfTeX3{
1761 \def\MT@scale@to@em{%
1762 \@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`.

```

1763 \MT@scale\@tempcntb \@tempb \MT@dimen@six
1764 \ifnum\@tempcntb=\z@ \else
1765 \MT@scale@factor
1766 \fi
1767 }

```

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

```

1768 \def\MT@get@charwd{%
1769 <*pdf- >
1770 ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1771 ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1772 ^^Q \MT@count=\wd\z@
1773 </pdf- >
1774 <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).

```

1775 <*xe- >
1776 \ifnum\MT@char@<\z@
1777 \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}%
1778 \MT@count=\wd\z@
1779 \else

```

```

1780 \MT@count=\fontcharwd\MT@font\MT@char@ \relax
1781 \fi
1782 </xe- >
1783 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1784 }

```

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 1em = \fontdimen 6.

```

1785 < *pdf- >
1786 \MT@requires@pdftex6{
1787 \g@addto@macro\MT@get@charwd{%
1788 \MT@ifdefined@c@T\MT@letterspace@
1789 {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1790 }
1791 } \relax
1792 }{

```

No adjustment with versions 0.14f and 0.14g.

```

1793 \def\MT@scale@to@em{%
1794 \MT@count=\@tempb \relax
1795 \ifnum\MT@count=\z@ \else
1796 \MT@scale@factor
1797 \fi
1798 }

```

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

```

1799 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1800 }
1801 </pdf- >
1802 </pdf-|lua-|xe- >
1803 </pdf-|lua-|xe-|show >

```

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

```

1804 < *package >
1805 \def\MT@get@font@dimen#1{%
1806 \ifnum\fontdimen#1\MT@font=\z@
1807 \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1808 \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1809 You should use a different `unit' for \MT@curr@list@name}%
1810 \else
1811 \MT@count=\fontdimen#1\MT@font
1812 \fi
1813 }

```

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

```

1814 \def\MT@info@missing@char{%
1815 \MT@info@n1{Character '\the\MT@toks'
1816 ^^X \ifnum\MT@char@<\z@ is missing\else
1817 ^^X \iffontchar\MT@font\MT@char@
1818 has a width of 0pt
1819 ^^X \else is missing\fi\fi
1820 ^^Q \MessageBreak (it's probably missing)
1821 \MessageBreak in font '\MT@font'. \MessageBreak
1822 Ignoring protrusion settings for this character}%
1823 }

```

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

```

1824 \def\MT@scale@factor{%
1825 \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1826 \expandafter\MT@scale\expandafter \@tempcntb
1827 \csname MT@\MT@feat @factor@\endcsname \@m
1828 \fi
1829 \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax

```

```

1830 \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1831 \else
1832 \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1833 \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1834 \fi
1835 \fi
1836 }

\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.

1837 \def\MT@warn@code@too@large#1{%
1838 \@tempcnta=#1\relax
1839 \ifnum\csname MT@\MT@feat @factor\endcsname=\@m \else
1840 \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1841 \@m \csname MT@\MT@feat @factor\endcsname
1842 \fi
1843 \MT@scale\@tempcnta \MT@dimen@six \MT@count
1844 \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@temp\space
1845 is too large for character\MessageBreak
1846 `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1847 Setting it to the maximum of \number\@tempcnta}%
1848 \@tempcntb=#1\relax
1849 }

\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.).

1850 \def\MT@get@opt{%
1851 \MT@set@listname

\MT@pr@factor@            Apply a factor?

\MT@sp@factor@            1852 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@            1853 \MT@let@nn{MT@\MT@feat @factor@}
                            {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
                            1854 \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
                            \number\csname MT@\MT@feat @factor\endcsname/1000}%
                            1855 }{%
                            1856 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
                            1857 }%
                            1858 }%
                            1859 }%

\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@              1860 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
                            1861 \MT@let@nn{MT@\MT@feat @unit@}%
                            1862 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
                            1863 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
                            1864 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                            relative to character widths}%
                            1865 \else
                            1866 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
                            1867 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                            relative to width of space}%
                            1868 \fi
                            1869 \fi
                            1870 }%
                            1871 \fi
                            1872 }%
                            1873 \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
                            1874 }%

\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.

1875 \let\MT@get@char@unit\relax

```

```

1876 \let\MT@get@space@unit@gobble
1877 \MT@exp@cs@ifx{MT@MT@feat @unit@}\@empty
1878 \let\MT@get@char@unit\MT@get@charwd
1879 \else
1880 \MT@exp@cs@ifx{MT@MT@feat @unit@}\m@ne
1881 \let\MT@get@space@unit\MT@get@font@dimen
1882 \else
1883 \MT@exp@cs\MT@get@unit{MT@MT@feat @unit@}%
1884 \fi
1885 \fi

```

Preset all characters? If so, we surely don't need to reset, too.

```

1886 \MT@ifdefined@n@T{MT@MT@feat @c@\csname MT@MT@feat @c@name\endcsname @preset}{%
1887 \csname MT@preset@MT@feat\endcsname
1888 \MT@let@nc{MT@reset@MT@feat @codes}\relax
1889 }%
1890 }

```

`\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.

```

1891 \def\MT@get@unit#1{%
1892 \expandafter\MT@get@unit@#1 e!\@nil
1893 \ifx\x\@empty\else\let#1\x\fi
1894 \@defaultunits\@tempdima#1 pt\relax\@nnil
1895 \ifdim\@tempdima=\z@
1896 \MT@warning@n1{%
1897 Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1898 width. Setting factors of list \@nameuse{MT@MT@feat @c@name}'\MessageBreak
1899 relative to character widths instead}%
1900 \let#1\@empty
1901 \let\MT@get@char@unit\MT@get@charwd
1902 \else
1903 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1904 to \the\@tempdima}%
1905 \MT@count=\@tempdima\relax
1906 \fi
1907 }
1908 \def\MT@get@unit@#1e#2#3\@nil{%
1909 \ifx\#3\@empty\else
1910 \if m#2%
1911 \edef\x{#1\fontdimen6\MT@font}%
1912 \else
1913 \if x#2%
1914 \edef\x{#1\fontdimen5\MT@font}%
1915 \fi
1916 \fi
1917 \fi
1918 }

```

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

```
1919 \def\MT@set@inputenc#1{%
```

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

```

1920 \def\MT@cat{#1}%
1921 \edef\@tempa{MT@MT@feat @#1\@csname MT@MT@feat @#1@name\endcsname @inputenc}%
1922 \MT@ifdefined@n@T\@tempa\MT@set@inputenc
1923 }

```

`\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.

```
1924 \MT@addto@setup{%
```



```

1925 \ifpackageloaded{inputenc}{%
1926 \ifpackagelater{inputenc}{2006/02/22}{%
1927 \def\MT@set@inputenc{%
1928 \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1929 \MT@load@inputenc
1930 }%
1931 }{%
1932 \let\MT@set@inputenc\MT@load@inputenc
1933 }%
1934 }{%
1935 \def\MT@set@inputenc{%
1936 \MT@warning@nl{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1937 \MessageBreak package isn't loaded. Ignoring input encoding}%
1938 }%
1939 }%
1940 }

```

`\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.

```

1941 \def\MT@load@inputenc{%
1942 \MT@cfg@catcodes
1943 <debug>\MT@dinfoln{1}{loading input encoding: \@nameuse{\@tempa}}%
1944 \inputencoding{\@nameuse{\@tempa}}%
1945 }

```

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

```

1946 \def\MT@set@pr@heirs#1{%
1947 \lcode\MT@font #1=\lcode\MT@font\MT@char\relax
1948 \rcode\MT@font #1=\rcode\MT@font\MT@char\relax
1949 <debug>\MT@dinfoln{2}{-- heir of \MT@char: #1}%
1950 <debug>\MT@dinfoln{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1951 <debug> \number\rcode\MT@font\MT@char}%
1952 }

```

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

```

1953 \def\MT@set@pr@prefixheirs{%
1954 \MT@ifdefined@c@T\MT@pr@inh@name{%
1955 \MT@ifdefined@nT{\MT@inh@\MT@pr@inh@name @prefixes}{%
1956 \MT@exp@cs\MT@map@tlist@c
1957 {\MT@inh@\MT@pr@inh@name @prefixes}%
1958 \MT@set@pr@prefixes
1959 }%
1960 }%
1961 }
1962 </package>

```

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

```

1963 <*pdf-|lua-|xe-|show>
1964 <pdf-|lua-|xe->\def\MT@set@pr@prefixes#1{\MT@set@pr@prefixes@#1}
1965 <pdf-|lua-|xe->\def\MT@set@pr@prefixes@#1#2#3#4%
1966 <show>\def\MTS@set@pr@prefixes@#1#2#3#4%
1967 {%
1968 <show> \MTS@lp@=\z@ \MTS@rp@=\z@
1969 <show> \ifnum#1=\@tempcntb \else
1970 <show> \par\leavevmode
1971 <show> \llap{\MTS@show@char@pr{#1} \MTS@printtext{=} }%
1972 <show> \fi
1973 <*xe->
1974 \edef\@tempa{\expandafter\ifx\@car#1\@nil U\@gobble#1\else\number\XeTeXglyphindex"#1" \fi}%
1975 \edef\@tempb{\expandafter\ifx\@car#2\@nil U\@gobble#2\else\number\XeTeXglyphindex"#2" \fi}%
1976 </xe->

```

```

1977 \@tempcnta=\z@
1978 \ifnum#3>\z@
1979   \@tempcnta=numexpr
1980 <pdf-|lua-|show> (\fontcharwd\MT@font#2-\fontcharwd\MT@font#1)%
1981 <xe-> (\fontcharwd\MT@font\@tempb-\fontcharwd\MT@font\@tempa)%
1982   *#3/\MT@dimen@six\relax
1983 \fi
1984 <pdf-|lua-|xe-> \lpcode\MT@font #2=numexpr\lpcode\MT@font#1+\@tempcnta\relax
1985 <show> \MTS@lp@=\dimexpr\numexpr\lpcode\MT@font#1+\@tempcnta\relax em/1000\relax
1986 \@tempcnta=\z@
1987 \ifnum#4>\z@
1988   \@tempcnta=numexpr
1989 <pdf-|lua-|show> (\fontcharwd\MT@font#2-\fontcharwd\MT@font#1)%
1990 <xe-> (\fontcharwd\MT@font\@tempb-\fontcharwd\MT@font\@tempa)%
1991   *#4/\MT@dimen@six\relax
1992 \fi
1993 <pdf-|lua-|xe-> \rprcode\MT@font #2=numexpr\rprcode\MT@font#1+\@tempcnta\relax
1994 <show> \MTS@rp@=\dimexpr\numexpr\rprcode\MT@font#1+\@tempcnta\relax em/1000\relax
1995 <debug>\MT@dinfoln{2}{-- (prefix) heir of #1: #2}%
1996 <debug>\MT@dinfoln{4}{;;; lp/rp (#2): \number\lpcode\MT@font#2/%
1997 <debug> \number\rprcode\MT@font#2}%
1998 <show> \MTS@show@char@pr{#2}%
1999 <show> \@tempcntb=#1\relax
2000 }
2001 </pdf-|lua-|xe-|show>

```

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

```

\MT@preset@pr@ 2002 <*package>
2003 \def\MT@preset@pr{%
2004   \expandafter\expandafter\expandafter\MT@preset@pr@
2005   \c@name MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
2006 }
2007 \def\MT@preset@pr@#1,#2\@nil{%
2008   \ifx\MT@pr@unit@\@empty
2009     \MT@warn@preset@t@width{pr}%
2010     \let\MT@preset@aux\MT@preset@aux@factor
2011   \else
2012     \def\MT@preset@aux{\MT@preset@aux@space2}%
2013   \fi
2014   \MT@i@fempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
2015   \MT@i@fempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
2016   \MT@set@all@pr\@tempa\@tempb
2017 }

```

`\MT@preset@aux` Auxiliary macro for presetting. Store value (#1) in macro (#2).

```

\MT@preset@aux@factor 2018 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 2019   \@tempcntb=#1\relax
2020   \MT@scale@factor
2021   \edef#2{\number\@tempcntb}%
2022 }
2023 \def\MT@preset@aux@space#1#2#3{%
2024   \def\@tempb{#2}%
2025   \MT@get@space@unit#1%
2026   \MT@scale@to@em
2027   \edef#3{\number\@tempcntb}%
2028 }

```

`\MT@warn@preset@t@width`

```

2029 \def\MT@warn@preset@t@width#1{%
2030   \MT@warning@n1{%
2031     Cannot preset characters relative to their widths\MessageBreak
2032     for \@nameuse{MT@abbr#1} list \@nameuse{MT@#1@c@name}'.
2033     Presetting them\MessageBreak relative to lem instead}%
2034 }

```

1.2.2 Manual protrusion

<code>\noprotusion</code>	This command may be used to inhibit protrusion on either side. It's part of L ^A T _E X since 2018-12-01. We provide it for older releases.
	<pre> 2035 \MT@ifdefined@c@TF\noprotusion\relax{ 2036 \DeclareRobustCommand\noprotusion{\leavevmode\kern-\p@\kern\p@} 2037 }</pre>
<code>\noprotusionifhmode</code>	Same, but only if we're already in hmode.
	<pre> 2038 \DeclareRobustCommand\noprotusionifhmode{\relax\ifhmode\kern-\p@\kern\p@\fi}</pre>
<code>\leftprotrusion</code>	This command may be used to add protrusion on the left hand side. We try to reconstruct the next glyph (possibly a ligature). ⁵
	<pre> 2039 \DeclareRobustCommand\leftprotrusion{% 2040 \MT@toks{ }% 2041 \MT@prot@toks{ }% 2042 \let\MT@prot@l\MT@prot@l@ 2043 \let\MT@prot@get@first@group\MT@prot@get@first@group@ 2044 \let\MT@maybe@textcmd\@firstofone 2045 \MT@prot@get@firstgroup 2046 }</pre>
<code>\MT@prot@l</code>	This probably doesn't need to be <code>\long</code> any longer.
<code>\MT@prot@l@</code>	<pre> 2047 \def\MT@prot@l@#1{% 2048 \MT@get@prot{#1}{left}% 2049 #1% 2050 }</pre>
<code>\MT@prot@toks</code>	If <code>\leftprotrusion</code> is followed by a text command, we trial-typeset only the first glyph, then actually typeset the whole argument, which we've saved in <code>\MT@prot@toks</code> , and finally gobble anything that might still be left in the input stream (see <code>\MT@prot@check@F</code> below).
<code>\MT@prot@l@etc</code>	
<code>\MT@gobble@to@nil</code>	<pre> 2051 \newtoks\MT@prot@toks 2052 \def\MT@prot@l@etc#1{% 2053 \MT@get@prot{\MT@maybe@textcmd{#1}}{left}% 2054 \the\MT@prot@toks 2055 \MT@gobble@to@nil 2056 } 2057 \def\MT@gobble@to@nil#1\MT@nil{ }</pre>
<code>\rightprotrusion</code>	Unfortunately, there's no way to retrieve anything that's already been typeset, so the counterpart cannot be defined symmetrically.
<code>\MT@prot@r</code>	<pre> 2058 \DeclareRobustCommand\rightprotrusion{\MT@prot@r} 2059 \def\MT@prot@r#1{% 2060 {#1}% 2061 \MT@get@prot{#1}{right}% 2062 }</pre>
<code>\MT@get@prot</code>	Typeset the text inside a box and get the left and right margin kerns. We add an extra <code>\vbox</code> in case we're inside a tabular. <code>\@newlistfalse</code> is meant to make <code>\</code> 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 ...). (We no longer reset counters etc., since we don't typeset groups anymore.)
<code>\MT@prot@hook</code>	Furthermore, we have a hook for compatibility fixes (currently used for <code>csquotes</code> only),

⁵ Lua_T_EX 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). Maybe *Marcel Krüger's* attempt at a `betterprotrusionboundary` (<https://tex.stackexchange.com/a/629080>) could be an option.

`\MT@csq@eqgroup` and a dedicated command to end `csquotes`'s group (because we actually typeset the quote character, instead of disabling quotes altogether (as we suggested for [issue #1], which was wrong)). Compatibility with `csquotes` is also the reason for the extra `\relax` after `<#1>`.

`\MT@noindent` Finally, L^AT_EX's new paragraph hooks require special attention, as they're (currently?) unable to distinguish between real typesetting and trial runs. In our case, fortunately, we really don't want to trigger the hooks.⁶ Also, as far as I can tell, we don't need a `\RawParEnd` at the end (as suggested in `ltpara`), because none of our commands are `\long` anymore.

```

2063 \let\MT@prot@hook\@empty
2064 \let\MT@csq@eqgroup\relax
2065 \IfFormatAtLeastTF{2021/11/15}
2066   {\let\MT@noindent\RawNoindent}
2067   {\let\MT@noindent\noindent}
2068 \def\MT@get@prot#1#2{%
2069   \begingroup
2070     \setbox\MT@tempbox\vbox{%
2071       \everypar{}%
2072       \parfillskip=\z@skip
2073       \hbadness\@M
2074       \clubpenalty\z@
2075       \widowpenalty\z@
2076       \interlinepenalty\z@
2077       \@newlistfalse
2078       \MT@prot@hook
2079       \MT@noindent #1\relax\MT@csq@eqgroup}%
2080     \vbadness=\@M
2081     \splittopskip=\z@
2082     \vfuzz=\maxdimen
2083     \setbox\MT@tempbox\vbox{%
2084       \ifvbox\MT@tempbox
2085         \global\setbox\MT@tempbox=\vsplit\MT@tempbox to \normalbaselineskip
2086         \unvbox\MT@tempbox
2087         \global\setbox\MT@tempbox=\lastbox
2088       \fi
2089     }%
2090   \endgroup
2091   \ifhbox\MT@tempbox
2092     \@tempdima=\@nameuse{#2margin kern}\MT@tempbox\relax
2093     \expandafter\ifdim\@tempdima=\z@ \else
2094       \leavevmode
2095       \MT@vinfo{<< adding #2 margin kern for `#1':\MessageBreak
2096         \the\@tempdima \oneline}%
2097       \kern\@tempdima
2098   <debug>%\vbox to0pt{\vss\llap{\fbox{%
2099   <debug>%       \MT@ifstreq{#2}{left}{\kern\@tempdima}\relax
2100   <debug>%       \kern-\fboxsep\unhbox\MT@tempbox\kern-\fboxsep
2101   <debug>%       \MT@ifstreq{#2}{right}{\kern\@tempdima}\relax}\hskip\marginparsep}}%
2102     \fi
2103   \fi
2104 }

```

`\MT@prot@ifx` Test next token.

```

2105 \def\MT@prot@ifx#1{%
2106   \ifx\MT@prot@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2107 }

```

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

```

2108 \def\MT@prot@ifcat#1{%

```

⁶ Well, in some cases we do, but this indeed 'needs further analysis' (cf. <https://github.com/latex3/latex2e/issues/880>).

```

2109 \ifcat#1\noexpand\MT@prot@next\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2110 }

\MT@prot@ifmacro      Test whether (#1) is a macro or an active character that does not take an argument.
\MT@prot@ifmacro@    As we're using etoolbox here, this only works with e-TeX.
2111 ^^X\def\MT@prot@ifmacro@{%
2112 ^^X \ifdefmacro\MT@prot@next{\ifdefparam\MT@prot@next\@gobble\@firstofone}\@gobble}
2113 ^^Q\let\MT@prot@ifmacro@gobble

\MT@prot@iffirstcmd  Test whether the first token in \MT@prot@next (once expanded) is the command
                      (#1). Since \MT@prot@next may also be user-defined (or whatever), we have to use
                      our own, \long version of \@car.
2114 \def\MT@prot@iffirstcmd#1{%
2115 \ifx\relax#1\expandafter\@secondoftwo\else
2116 \MT@exp@two@c\ifx\MT@car\MT@prot@next\relax\@nil#1%
2117 \expandafter\expandafter\expandafter\@firstoftwo
2118 \else
2119 \expandafter\expandafter\expandafter\@secondoftwo
2120 \fi
2121 \fi
2122 }

\MT@car              A long car.
2123 \long\def\MT@car#1#2\@nil{#1}

\MT@prot@iflicrcmd  Fun with LICR: If we have an encoding command, test if the first command of the
                      third command (e.g., \T1\") is \@text@composite, in which case also grab the next
                      token, otherwise it should be a text command.
2124 \def\MT@getthird#1#2#3#4\@nil{#3}
2125 \def\MT@prot@iflicrcmd{%
2126 \MT@prot@iffirstcmd\@current@cmd\@secondoftwo\@firstofone
2127 {\MT@prot@iffirstcmd\@changed@cmd\@firstofone\@gobble}%
2128 {\expandafter\expandafter\expandafter\let
2129 \expandafter\expandafter\expandafter\@tempa
2130 \expandafter\MT@getthird\MT@prot@next\relax\@nil
2131 \MT@exp@two@c\ifx\@car\@tempa\relax\@nil\@text@composite
2132 \def\MT@temp*##1##2{\MT@exp@one@n\MT@prot@1{\the\MT@toks##1##2}}%
2133 \else
2134 \def\MT@temp*##1{\MT@exp@one@n\MT@prot@1{\the\MT@toks##1}}%
2135 \fi
2136 }%
2137 }

\MT@prot@addgroup   If we have a group, we inject \MT@prot@get@firstgroup again at the beginning and
                      don't bother about the rest. This still allows, e.g., \verb, verbatim or lstlistings
                      material. The downside of being this cautious is that we'll miss lots of cases.
2138 \def\MT@prot@addgroup{\bgroup\afterassignment\MT@prot@get@firstgroup\let\MT@temp= }

\MT@prot@get@firstgroup  Scan token by token.
\MT@prot@get@firstgroup@tc 2139 \def\MT@prot@get@firstgroup{\futurelet\MT@prot@next\MT@prot@get@first@group}
\MT@prot@get@firststoken 2140 \def\MT@prot@get@firstgroup@tc{\futurelet\MT@prot@next\MT@prot@get@first@group@tc}
\MT@prot@get@nexttoken 2141 \def\MT@prot@get@firststoken{\futurelet\MT@prot@next\MT@prot@get@first@token}
2142 \def\MT@prot@get@nexttoken{\futurelet\MT@prot@next\MT@prot@get@next@token}

\MT@prot@check      We map through a list of commands that should be copied into the toks. (#3) will
\MT@prot@check@    be \relax by default, but can also indicate a replacement command.
2143 \def\MT@prot@check#1{\MT@prot@check@#1\relax\@nil}
2144 \def\MT@prot@check@#1#2#3\@nil{%
2145 \ifx\MT@prot@next#2%
2146 \csname MT@prot@check@#1\endcsname #3%
2147 \let\MT@prot@ifmacro@gobble
2148 \expandafter\MT@tlist@break

```

```
2149 \fi
2150 }
```

Beware that the following nomenclature is rather arcane.

`\MT@prot@check@I` • This is for commands to be Ignored.

```
2151 \def\MT@prot@check@I{%
2152 \def\MT@temp*##1{\MT@prot@get@firstgroup}%
2153 }
```

`\MT@prot@check@S` • Add a Single command (without an argument).

```
2154 \def\MT@prot@check@S{%
2155 \def\MT@temp*##1{\MT@toks\expandafter{\the\MT@toks##1}\MT@prot@get@firstgroup}%
2156 }
```

`\MT@prot@check@X` • Add a command with One argument.

```
2157 \def\MT@prot@check@X{%
2158 \def\MT@temp*##1##2{\MT@toks\expandafter{\the\MT@toks##1{##2}}\MT@prot@get@firstgroup}%
2159 }
```

`\MT@prot@check@T` • Add a command with Two arguments.

```
2160 \def\MT@prot@check@T{%
2161 \def\MT@temp*##1##2##3{\MT@toks\expandafter{\the\MT@toks##1{##2}{##3}}\MT@prot@get@firstgroup}%
2162 }
```

`\MT@prot@check@E` • This is for commands that Enclose their argument in something, e.g., in braces, and which we trial-typeset without any contents.

```
2163 \def\MT@prot@check@E{%
2164 \the\MT@toks
2165 \def\MT@temp*##1{\MT@prot@1{##1}}%
2166 }
```

`\MT@prot@check@e` • Same for starred commands (the main candidate here is `csquotes's \enquote`).

```
2167 \def\MT@prot@check@e{%
2168 \the\MT@toks
2169 \def\MT@temp*##1{\@ifstar{\MT@prot@1{##1*}}{\MT@prot@1{##1}}}%
2170 }
```

`\MT@prot@check@eX` • Here we replace the ‘integrated interface’ (`csquotes`) with the regular one.

```
2171 \def\MT@prot@check@eX#1{%
2172 \the\MT@toks
2173 \def\MT@temp*##1{\@ifstar
2174 {\MT@get@prot{##1*}{left}##1*}
2175 {\MT@get@prot{##1}{left}##1}}%
2176 }
```

`\MT@prot@check@l` • `csquotes` provides a couple of commands for quotations in foreign languages (lowercase, because it may be starred), whose first argument (the language) we also have to evaluate before trial typesetting.

```
2177 \def\MT@prot@check@l{%
2178 \def\MT@temp*##1{\@ifstar{\MT@prot@check@l@##1*}{\MT@prot@check@l@##1}}%
2179 }
2180 \def\MT@prot@check@l@#1#2{%
2181 \the\MT@toks
2182 \MT@prot@l{##1}{##2}}%
2183 }
```

`\MT@prot@check@1X` • Another macro for csquotes commands: replace integrated language-switching commands with their regular variants.

```
2184 \def\MT@prot@check@1X#1{%
2185   \def\MT@temp*##1{\@ifstar
2186     {\def\MT@temp{##1*}\MT@prot@check@1X@{#1*}}
2187     {\def\MT@temp{##1}\MT@prot@check@1X@{#1}}}%
2188 }
2189 \def\MT@prot@check@1X@#1#2{%
2190   \the\MT@toks
2191   \MT@get@prot{#1{#2}}{\left}\MT@temp{#2}%
2192 }
```

`\MT@prot@check@F` • Here we deal with Font switching commands (i.e., text commands, which take an argument). We (a) remember the text command, (b) save the full text, and then (c) continue inspecting the contents of the argument. We also have to execute (and empty) `\MT@toks`, because it might already contain other commands. Nested text commands still don't work.

```
2193 \def\MT@prot@check@F{%
2194   \ifx\MT@prot@1\MT@prot@1@etc
2195     \def\MT@temp*{\MT@exp@one@n\MT@prot@1{\the\MT@toks}}%
2196   \else
2197     \let\MT@prot@1\MT@prot@1@etc
2198     \let\MT@prot@get@first@group\MT@prot@get@first@group@etc
2199     \def\MT@temp*##1##2{%
2200       \let\MT@maybe@textcmd##1%
2201       \the\MT@toks
2202       \MT@toks{}%
2203       \MT@prot@toks{##1{##2}}%
2204       \MT@prot@get@first@group@etc##2\MT@ni
2205     }%
2206   \fi
2207 }
```

`\MT@prot@check@C` • Same, but for commands that allow an optional argument (e.g., the Case changing commands since L^AT_EX 2022/11/01).

```
2208 \def\MT@prot@check@C{%
2209   \ifx\MT@prot@1\MT@prot@1@etc
2210     \def\MT@temp*{\MT@exp@one@n\MT@prot@1{\the\MT@toks}}%
2211   \else
2212     \let\MT@prot@1\MT@prot@1@etc
2213     \let\MT@prot@get@first@group\MT@prot@get@first@group@etc
2214     \def\MT@temp*##1{%
2215       \the\MT@toks
2216       \MT@toks{}%
2217       \@ifnextchar[%
2218         {\MT@prot@check@C@##1}%
2219         {\MT@prot@check@C@##1[]}%
2220     }%
2221   \fi
2222 }
2223 \def\MT@prot@check@C@#1[#2]#3{%
2224   \def\MT@maybe@textcmd{#1[#2]}%
2225   \MT@prot@toks{#1[#2]{#3}}%
2226   \MT@prot@get@first@group@tc#3\MT@ni
2227 }
```

`\MT@prot@check@cmds` And here's the list of commands that we can deal with.

```
2228 \def\MT@prot@check@cmds{%
2229   {I\ignorespaces}{I\relax}{I\empty}%
2230   {S\rmfamily}{S\sfamily}{S\ttfamily}{S\mdseries}{S\bfseries}%

```

```

2231 {\upshape}{\itshape}{\slshape}{\scshape}{\em}%
2232 {\normalfont}{\selectfont}%
2233 {\slsstyle}%
2234 {\tiny}{\scriptsize}{\footnotesize}{\small}{\normalsize}%
2235 {\large}{\Large}{\LARGE}{\huge}{\Huge}%
2236 {0\fontencoding}{0\fontfamily}{0\fontseries}{0\fontshape}%
2237 {0\microtypesetup}{0\microtypecontext}%
2238 {T\fontsize}%
2239 {F\textrm}{F\textsf}{F\texttt}{F\textnormal}%
2240 {F\textbf}{F\textmd}{F\textit}{F\textsl}{F\textsc}{F\textup}{F\emph}%
2241 }

```

L^AT_EX 2020/02/02 introduced some more text commands (adopted from fontaxes, which provides some more, see below).

```

2242 \IfFormatAtLeastTF{2020/02/02}
2243   {\g@addto@macro\MT@prot@check@cmds{%
2244     {\swshape}{\ulcshape}{\sscshape}{\normalshape}%
2245     {F\textulc}{F\textsw}{F\textssc}%
2246     {0\fontseriesforce}{0\fontshapeforce}}%
2247   \relax
2248 \IfFormatAtLeastTF{2022/11/01}
2249   {\g@addto@macro\MT@prot@check@cmds{{C\MakeUppercase}{C\MakeLowercase}{C\MakeTitlecase}}%
2250   {\g@addto@macro\MT@prot@check@cmds{{F\MakeUppercase}{F\MakeLowercase}}%

```

The ltxdoc class and the doc package provide some abbreviations. Unfortunately, the \cmd command doesn't work.

```

2251 \@ifclassloaded{ltxdoc}
2252   {\g@addto@macro\MT@prot@check@cmds{{E\enquote}{E\marg}{E\oarg}{E\parg}{E\cs}}\relax
2253 \MT@addto@setup{%
2254 \MT@with@package@T{doc}
2255   {\g@addto@macro\MT@prot@check@cmds{{E\meta}}}%

```

The additional fontaxes commands.

```

2256 \MT@with@package@T{fontaxes}
2257   {\g@addto@macro\MT@prot@check@cmds{%
2258     {\txfigures}{\lnfigures}{\tbfigures}{\prfigures}%
2259     {0\fontfigurestyle}{0\fontfigurealignment}{0\fontbasefamily}%
2260     {0\figureversion}%
2261     {F\textsw}{F\textssc}{F\textulc}%
2262     {F\textfigures}{F\liningfigures}{F\tabularfigures}{F\proportionalfigures}}%
2263 \IfFormatAtLeastTF{2020/02/02}\relax
2264   {\g@addto@macro\MT@prot@check@cmds{%
2265     {\swshape}{\ulcshape}{\sscshape}%
2266     {F\textulc}{F\textsw}{F\textssc}}%

```

The nfssect-cfr package (an extension of the nfssect package, which is part of Philipp Lehman's font installation guide but was never publicised separately as far as I can tell) adds many more commands on top of the NFSS.

```

2267 \MT@with@package@T{nfssect-cfr}
2268   {\g@addto@macro\MT@prot@check@cmds{%
2269     {\Ttistyle}{\ltistyle}{\ofstyle}{\altstyle}{\regstyle}{\embossstyle}%
2270     {\ornamentalstyle}{\qtstyle}{\shstyle}{\tmstyle}{\tvstyle}{\swashstyle}%
2271     {\lnstyle}{\osstyle}{\instyle}{\sustyle}{\lstyle}{\ostyle}%
2272     {\pstyle}{\tstyle}{\plstyle}{\postyle}{\tlstyle}{\tostyle}%
2273     {\scolshape}{\olshape}{\sishape}{\ushape}{\scushape}%
2274     {\uishape}{\rishape}{\dfshape}{\swstyle}%
2275     {\nwidth}{\cdwidth}{\ecwidth}{\ucwidth}%
2276     {\etwidth}{\epwidth}{\exwidth}{\uxwidth}{\regwidth}%
2277     {\mbweight}{\dbweight}{\sbweight}{\ebweight}%
2278     {\subweight}{\lgweight}{\leweight}{\ulweight}%
2279     {F\textti}{F\textlt}{F\textof}{F\textalt}{F\textreg}{F\textreg}{F\emboss}%
2280     {F\textorn}{0\ornament}{F\textqt}{F\textsh}{F\texttm}{F\texttv}{F\textswash}%
2281     {F\textln}{F\textos}{F\textin}{F\textsu}{F\textl}{F\texto}%
2282     {F\textp}{F\textt}{F\textpl}{F\textpo}{F\texttl}{F\textto}%

```



```

2283     {F\textol}{F\textsi}{F\textu}{F\textscu}%
2284     {F\textui}{F\textri}{F\textdf}%
2285     {F\textnw}{F\textcd}{F\textec}{F\textuc}%
2286     {F\textet}{F\textep}{F\textex}{F\textux}{F\textrw}%
2287     {F\textmb}{F\textdb}{F\textsb}{F\texteb}%
2288     {F\textub}{F\textlg}{F\textel}{F\textul}}%
2289     \IfFormatAtLeastTF{2020/02/02}\relax
2290     {\g@addto@macro\MT@prot@check@cmds{{S\swshape}{F\textsw}}}}

```

If `yfonts` is loaded, we add the relevant commands.

```

2291 \MT@with@package@T{yfonts}
2292   {\g@addto@macro\MT@prot@check@cmds{%
2293     {S\frakfamily}{S\swabfamily}{S\gothfamily}%
2294     {F\textfrak}{F\textswab}{F\textgoth}}}%

```

`csquotes`'s `\enquote` command. It would take precedence over the one provided by `ltxdoc`.

```

2295 \MT@with@package@T{csquotes}
2296   {\ifclassloaded{ltxdoc}
2297     {\patchcmd\MT@prot@check@cmds{E\enquote}{e\enquote}\relax\relax}
2298     {\g@addto@macro\MT@prot@check@cmds{{e\enquote}}}%
2299     \g@addto@macro\MT@prot@check@cmds{{e\textquote}%
2300     {\lforeignquote}{\lhyphenquote}{\lforeigntextquote}{\lhyphentextquote}%
2301     {{eX}\textcquote\textquote}%
2302     {{\lX}\foreigntextcquote\foreigntextquote}%
2303     {{\lX}\hyphentextcquote\hyphentextquote}}}%
2304 }

```

`\MT@prot@get@first@group` If next char is `{`, start a group and try again, else continue until we find a beginning char.

```

2305 \def\MT@prot@get@first@group@{%
2306   \MT@prot@ifcat\bgroup{%
2307     \def\MT@temp*{\MT@prot@addgroup}%
2308   }{%
2309     \def\MT@temp*{\MT@prot@get@first@token}%
2310   }%
2311   \MT@temp*%
2312 }

```

`\MT@prot@get@first@group@tc` The variant for text commands (in case they start with another group).

```

2313 \def\MT@prot@get@first@group@tc{%
2314   \MT@prot@ifcat\bgroup{%
2315     \def\MT@temp*##1##2\MT@nil{\MT@ifempty{##1}\relax
2316     {{\MT@prot@get@first@group@tc##1\MT@nil}}}%
2317   }{%
2318     \def\MT@temp*{\MT@prot@get@first@token}%
2319   }%
2320   \MT@temp*%
2321 }

```

`\MT@prot@get@first@token` This can be called repeatedly. We add a letter or other character, ...

```

2322 \def\MT@prot@get@first@token{%
2323   \def\MT@temp*{\MT@exp@one@n\MT@ifempty{\the\MT@toks}
2324     {\MT@exp@one@n\MT@ifempty{\the\MT@prot@toks}\relax{\the\MT@prot@toks\MT@gobble@to@nil}}
2325     {\MT@exp@one@n\MT@prot@1{\the\MT@toks}}}%
2326   \MT@prot@ifcat{a}{%
2327     \def\MT@temp*{\MT@prot@addtoken@first}%
2328   }{%
2329     \MT@prot@ifcat{!}{%
2330       \def\MT@temp*{\MT@prot@addtoken@first}%
2331     }%

```

a space character, ...

```

2332   \MT@prot@ifx\@sptoken{%

```

```
2333     \def\MT@temp* {\MT@prot@get@firstgroup}%
2334     }{%
```

commands, ...

```
2335     \let\MT@prot@ifmacro\MT@prot@ifmacro@
2336     \MT@map@tlist@c\MT@prot@check@cmds\MT@prot@check
```

... or a command/active char whose first command is one of the following:

```
2337     \MT@prot@ifmacro{%
2338     \MT@prot@iffirstcmd\UTFvii@two@octets{%
2339     \def\MT@temp*##1##2{\MT@exp@one@n\MT@prot@1{\the\MT@toks##1##2}}%
2340     }{%
2341     \MT@prot@iffirstcmd\UTFvii@three@octets{%
2342     \def\MT@temp*##1##2##3{\MT@exp@one@n\MT@prot@1{\the\MT@toks##1##2##3}}%
2343     }{%
2344     \MT@prot@iffirstcmd\UTFvii@four@octets{%
2345     \def\MT@temp*##1##2##3##4{\MT@exp@one@n\MT@prot@1{\the\MT@toks##1##2##3##4}}%
2346     }%
```

(this is for chars made active by csquotes, via `\MakeAutoQuote` or `\MakeOuterQuote`)

```
2347     \MT@prot@iffirstcmd\csqq{\def\MT@temp*##1{\MT@exp@one@n\MT@prot@1{\the\MT@toks##1}}}%
```

or, finally, a LICR command.

```
2348     \MT@prot@iflicrcmd
2349     }%
2350     }%
2351     }%
2352     }%
2353     }%
2354     }%
2355     }%
2356     }%
2357     \MT@temp*%
2358 }
```

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

```
2359 \def\MT@prot@addtoken@first#1{%
2360   \MT@toks\expandafter{\the\MT@toks#1}%
2361   \MT@prot@get@nexttoken
2362 }
```

`\MT@prot@get@next@token` **Continue if letter or other.**

```
2363 \def\MT@prot@get@next@token{%
2364   \def\MT@temp*{\MT@prot@addtoken@next}%
2365   \MT@prot@ifcat{a}\relax{%
2366     \MT@prot@ifcat{!}\relax{%
2367       \def\MT@temp*{\MT@exp@one@n\MT@prot@1{\the\MT@toks}}%
2368     }%
2369   }%
2370   \MT@temp*%
2371 }
2372 /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.

```
2373 <*pdf-|lua-|xe-
2374 \def\MT@prot@addtoken@next#1{%
2375   \MT@toks\expandafter{\the\MT@toks#1}%
2376   \setbox\MT@tempbox\hbox{%
```

We disable italic correction, which would prevent us from seeing the ligature (with text commands).

```
2377   \let\maybe@ic\relax
```

```

2378 \MT@exp@one@n\MT@maybe@textcmd{\the\MT@toks}%
2379 <pdf-|xe- > \relax
2380 <lua- > }\setbox\MT@tempbox\hbox{\unhbox\MT@tempbox
2381 \ifnum\lastnodetype=7 \aftergroup\firstoftwo\else\aftergroup\secondoftwo\fi}%
2382 \MT@prot@get@nexttoken
2383 {\MT@exp@one@n\MT@prot@1{\the\MT@toks}}%
2384 }
2385 </pdf-|lua-|xe- >

```

1.2.3 Expansion

`\MT@expansion` Set up for expansion?

```

2386 <*pdf-|lua- >
2387 \def\MT@expansion{\MT@maybe@do{ex}}

```

`\MT@set@ex@codes@e` 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).

```

2388 \def\MT@set@ex@codes@e{%
2389 \MT@if@list@exists{%
2390 \MT@get@ex@opt
2391 \let\MT@get@char@unit\relax
2392 \MT@reset@ef@codes
2393 \MT@get@inh@list
2394 \MT@set@inputenc{c}%
2395 \MT@load@list\MT@ex@cc@name
2396 \MT@set@listname
2397 \MT@let@cn\@tempc{\MT@ex@cc@\MT@ex@cc@name}%
2398 \expandafter\MT@set@codes\@tempc,\relax,%
2399 \MT@expandfont
2400 }\relax
2401 }
2402 </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.

```

2403 <package >\newif\ifMT@nonselected
2404 <*pdf-|lua- >
2405 \def\MT@set@ex@codes@n{%
2406 \MT@nonselectedtrue
2407 \MT@if@list@exists
2408 \MT@get@ex@opt
2409 {%
2410 \let\MT@stretch@ \MT@stretch
2411 \let\MT@shrink@ \MT@shrink
2412 \let\MT@step@ \MT@step
2413 \let\MT@auto@ \MT@auto
2414 \let\MT@ex@factor@ \MT@ex@factor
2415 }%
2416 \MT@reset@ef@codes
2417 \MT@expandfont
2418 \MT@nonselectedfalse
2419 }

```

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

```

2420 \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.

```

2421 <lua-
2422 \MT@requires@luatex3{
2423 \MT@requires@luatex4{\let\pdffontexpand\expandglyphsinfont}\relax
2424 \ifnum\luatexversion<79
2425 \def\MT@expandfont{%
2426 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ autoexpand\relax
2427 }
2428 \else
2429 \def\MT@expandfont{%
2430 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@\relax
2431 }
2432 \fi
2433 }{
2434 </lua-
2435 \def\MT@expandfont{%
2436 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
2437 }
2438 </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).

```

2439 \def\MT@set@all@ex#1{%
2440 <debug>\MT@dinfol{3}{-- ex: setting all to \number#1}%
2441 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
2442 }
2443 \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).

```

2444 <pdf->\MT@requires@pdfTeX4
2445 <lua->\MT@requires@luatex5
2446 {
2447 \def\MT@reset@ef@codes{%
2448 \ifnum\MT@ex@factor@=\@m \else
2449 \MT@reset@ef@codes@
2450 \fi
2451 }
2452 }{
2453 \let\MT@reset@ef@codes\MT@reset@ef@codes@
2454 }

```

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

```

2455 \def\MT@ex@split@val#1\relax{%
2456 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

2457 \ifnum\MT@ex@factor@=\@m \else
2458 \MT@scale\@tempcntb \MT@ex@factor@ \@m
2459 \fi
2460 \ifnum\@tempcntb > \MT@ex@max
2461 \MT@warn@ex@too@large\MT@ex@max
2462 \else
2463 \ifnum\@tempcntb < \MT@ex@min
2464 \MT@warn@ex@too@large\MT@ex@min
2465 \fi
2466 \fi
2467 \efcode\MT@font\MT@char=\@tempcntb
2468 <debug>\MT@dinfol{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

2469 \MT@ifdefined@c@T\MT@ex@inh@name{%
2470 \MT@ifdefined@n@T{MT@inh@MT@ex@inh@name @\MT@char @}{%
2471 \MT@exp@cs\MT@map@tlist@c{MT@inh@MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
2472 }%
2473 }%
2474 }

```

\MT@warn@ex@too@large

```

2475 \def\MT@warn@ex@too@large#1{%
2476 \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
2477 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
2478 Setting it to the maximum of \number#1}%
2479 \@tempcntb=#1\relax
2480 }

```

\MT@get@ex@opt Apply different values to this font?

```

\MT@ex@factor@ 2481 \def\MT@get@ex@opt{%
\MT@stretch@ 2482 \MT@set@listname
2483 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 2484 \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 2485 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 2486 }{%
2487 \let\MT@ex@factor@\MT@ex@factor
2488 }%
2489 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
2490 \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
2491 \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
2492 <lua- \MT@requires@luatex3\relax{%
2493 \MT@get@ex@opt@{auto}{\MT@ifstreql\MT@auto@}{autoexpand}{En}{Dis}abling automatic expansion}%
2494 <lua- }%
2495 \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}{%
2496 \MT@preset@ex
2497 \let\MT@reset@ef@codes\relax
2498 }%
2499 }

```

\MT@get@ex@opt@

```

2500 \def\MT@get@ex@opt@#1#2{%
2501 \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
2502 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
2503 \MT@vinfo{... : #2}%
2504 }{%
2505 \MT@let@nn{MT@#1@}{MT@#1}%
2506 }%
2507 }

```

\MT@set@ex@heirs

```

2508 \def\MT@set@ex@heirs#1{%
2509 \efcode\MT@font#1=\efcode\MT@font\MT@char
2510 <debug>\MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
2511 <debug>\MT@dinfo@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
2512 }

```

\MT@preset@ex

```

2513 \def\MT@preset@ex{%
2514 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
2515 \MT@scale@factor
2516 \MT@set@all@ex\@tempcntb
2517 }
2518 </pdf-|lua-

```

1.2.4 Interword spacing (glue)

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

```
2519 <pdf-
2520 \MT@requires@pdftex6{
2521 \def\MT@spacing{\MT@maybe@do{sp}}
```

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

```
2522 \def\MT@set@sp@codes{%
2523 \MT@if@list@exists{%
2524 \MT@get@opt
2525 \MT@reset@sp@codes
2526 \MT@get@inh@list
2527 \MT@set@inputenc{c}%
2528 \MT@load@list\MT@sp@cc@name
2529 \MT@set@listname
2530 \MT@let@cn\@tempc\MT@sp@c@\MT@sp@c@name}%
2531 \expandafter\MT@set@codes\@tempc,\relax,%
2532 }\MT@reset@sp@codes
2533 }
```

`\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).

```
2534 \def\MT@sp@split@val#1,#2,#3\relax{%
2535 \def\@tempb{#1}%
2536 \MT@if@empty\@tempb\relax{%
2537 \MT@get@space@unit2%
2538 \MT@scale@to@em
2539 \knbscode\MT@font\MT@char=\@tempcntb
2540 <debug>\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
2541 }%
2542 \def\@tempb{#2}%
2543 \MT@if@empty\@tempb\relax{%
2544 \MT@get@space@unit3%
2545 \MT@scale@to@em
2546 \stbscode\MT@font\MT@char=\@tempcntb
2547 <debug>\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
2548 }%
2549 \def\@tempb{#3}%
2550 \MT@if@empty\@tempb\relax{%
2551 \MT@get@space@unit4%
2552 \MT@scale@to@em
2553 \shbscode\MT@font\MT@char=\@tempcntb
2554 <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
2555 }%
2556 \MT@if@defined@c@T\MT@sp@inh@name{%
2557 \MT@if@defined@nT\MT@inh@\MT@sp@inh@name @\MT@char @}%
2558 \MT@exp@cs\MT@map@tlist@c\MT@inh@\MT@sp@inh@name @\MT@char @\MT@set@sp@heirs
2559 }%
2560 }%
2561 }
```

`\MT@set@sp@heirs`

```
2562 \def\MT@set@sp@heirs#1{%
2563 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
2564 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
2565 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
2566 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
2567 <debug>\MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
2568 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
2569 }
```

`\MT@set@all@sp`

`\MT@reset@sp@codes` 2570 \def\MT@set@all@sp#1#2#3{%

`\MT@reset@sp@codes@`

```

2571 (debug)\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
2572 \let\MT@temp@empty
2573 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
2574 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
2575 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
2576 \MT@do@font\MT@temp
2577 }
2578 \def\MT@reset@sp@codes@{\MT@set@all@sp@z@z@z@}
2579 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 2580 \def\MT@preset@sp{%
2581 \expandafter\expandafter\expandafter\MT@preset@sp@
2582 \csname MT@sp@code\MT@sp@code@name @preset\endcsname\@nil
2583 }
2584 \def\MT@preset@sp@#1,#2,#3\@nil{%
2585 \ifx\MT@sp@unit@\@empty
2586 \MT@warn@preset@t@width{sp}%
2587 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
2588 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
2589 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
2590 \else
2591 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
2592 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
2593 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
2594 \fi
2595 \MT@set@all@sp@\@tempa\@tempc\@tempb
2596 }
2597 }\relax

```

1.2.5 Additional kerning

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

```

2598 \MT@requires@pdftex6{
2599 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

2600 \def\MT@set@kn@codes{%
2601 \MT@if@list@exists{%
2602 \MT@get@opt
2603 \MT@reset@kn@codes
2604 \MT@get@inh@list
2605 \MT@set@inputenc{c}%
2606 \MT@load@list\MT@kn@code@name
2607 \MT@set@listname
2608 \MT@let@cn\@tempc\MT@kn@code\MT@kn@code@name}%
2609 \expandafter\MT@set@codes\@tempc,\relax,%
2610 }\MT@reset@kn@codes
2611 }

```

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

```

2612 \def\MT@kn@split@val#1,#2\relax{%
2613 \def\@tempb{#1}%
2614 \MT@ifempty\@tempb\relax{%
2615 \MT@get@space@unit2%
2616 \MT@scale@to@em
2617 \knbcode\MT@font\MT@char=\@tempcntb
2618 (debug)\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
2619 }%
2620 \def\@tempb{#2}%
2621 \MT@ifempty\@tempb\relax{%
2622 \MT@get@space@unit2%
2623 \MT@scale@to@em
2624 \knaccode\MT@font\MT@char=\@tempcntb

```

```

2625 (debug)\MT@dinfol{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
2626 }%
2627 \MT@ifdefined@cT\MT@kn@inh@name{%
2628 \MT@ifdefined@nT\MT@inh@MT@kn@inh@name @\MT@char @}{%
2629 \MT@exp@cs\MT@map@tlist@c\MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
2630 }%
2631 }%
2632 }

\MT@set@kn@heirs

2633 \def\MT@set@kn@heirs#1{%
2634 \knbccode\MT@font#1=\knbccode\MT@font\MT@char
2635 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
2636 (debug)\MT@dinfol{2}{-- heir of \MT@char: #1}%
2637 (debug)\MT@dinfol{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
2638 (debug) \number\knaccode\MT@font\MT@char}%
2639 }

\MT@set@all@kn

\MT@reset@kn@codes 2640 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 2641 (debug)\MT@dinfol{3}{-- knac/knbc: setting all to #1/#2}%
2642 \let\MT@temp@empty
2643 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbccode\MT@font\@tempcnta=#1\relax}}%
2644 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
2645 \MT@do@font\MT@temp
2646 }
2647 \def\MT@reset@kn@codes@\MT@set@all@kn\z@\z@
2648 \let\MT@reset@kn@codes\relax

\MT@preset@kn

\MT@preset@kn@ 2649 \def\MT@preset@kn{%
2650 \expandafter\expandafter\expandafter\MT@preset@kn@
2651 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
2652 }
2653 \def\MT@preset@kn@#1,#2\@nil{%
2654 \ifx\MT@kn@unit@\@empty
2655 \MT@warn@preset@twidth{kn}%
2656 \let\MT@preset@aux\MT@preset@aux@factor
2657 \else
2658 \def\MT@preset@aux{\MT@preset@aux@space2}%
2659 \fi
2660 \MT@ifempty{#1}\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
2661 \MT@ifempty{#2}\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
2662 \MT@set@all@kn\@tempa\@tempb
2663 }
2664 }\relax
2665 (pdf-)

```

1.2.6 Tracking

This only works with pdfTeX 1.40 or LuaTeX 0.62.

```

2666 (pdf- |lua-)
2667 (pdf-)\MT@requires@pdftex6
2668 (lua-)\MT@requires@luatex3
2669 {

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

\MT@tr@font@list 2670 \let\MT@tr@font@list\@empty
2671 \def\MT@tracking@{%
2672 \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
2673 \ifMT@in@list\@else
2674 \MT@maybe@do{tr}%

```



```

2675 \ifMT@do\else
2676 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
2677 \fi
2678 \fi
2679 }
2680 </pdf-|lua-|
2681 <pdf-|lua-|letterspace>\let\MT@tracking
2682 <pdf-|lua-| \MT@tracking@
2683 <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).

```

2684 <*pdf-|lua-|letterspace>
2685 \def\MT@set@tr@codes{%
2686 <*pdf-|lua-|
2687 \MT@vinfo{Tracking font `\'MT@@font'\on@line}%
2688 <*pdf-|
2689 \MT@requires@pdftex8\@firstofone{%
2690 \MT@ifdefined@TF{\MT@@font-fake6}{%
2691 \MT@exp@cs\ifx{\MT@@font-fake6}\@empty
2692 \MT@warning@nl{%
2693 Font `\'MT@@font' does not specify its\MessageBreak
2694 \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
2695 tracking will not work with this font}%
2696 \MT@gl@et@nc{\MT@@font-fake6}\relax
2697 \fi
2698 }%
2699 }%
2700 </pdf-|
2701 \MT@if@list@exists
2702 \MT@get@tr@opt
2703 \relax
2704 </pdf-|lua-|
2705 \MT@ifdefined@c@TF\MT@letterspace@relax{\let\MT@letterspace@\'MT@letterspace}%
2706 \ifnum\MT@letterspace@=\z@

```

Zero tracking requires special treatment.

```

2707 \MT@set@tr@zero
2708 \else
2709 <pdf-|lua-| \MT@vinfo{... Tracking by \number\MT@letterspace}%

```

Letterspacing only works in PDF mode.

```

2710 \MT@warn@tracking@DVI

```

`\MT@lsfont` The letterspaced font instances are saved in macros `\/<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.

```

2711 \xdef\MT@lsfont{\csname\expandafter\string\font@name
2712 \number\MT@letterspace@ls\endcsname}%
2713 \expandafter\ifx\MT@lsfont\relax
2714 <debug>\MT@dinfo@nl{1}{... new letterspacing instance}%

```

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

```

2715 \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`.

```

2716 (*lua-|letterspace)
2717     \MT@if@luaotf@font{%
2718 (lua-&debug)\MT@dinfo@n1{1}{... luaotf font: \MessageBreak
2719 (lua-&debug)         \expandafter\fontname\font@name}%
2720     \global\expandafter\font\MT@lsfont=\MT@lsfontspec@font
2721     }{%
2722 (lua-|letterspace)
2723 (lua-&debug)\MT@dinfo@n1{1}{... legacy font}%
2724     \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
2725 (lua-|letterspace)     }%

```

Scale interword spacing (not configurable in `letterspace`).

```

2726 (*pdf-|lua-)
2727     \MT@ifdefined@c@TF\MT@tr@ispace
2728     {\let\@tempa\MT@tr@ispace}%
2729     {\edef\@tempa{\MT@letterspace@*,,}}%
2730     \MT@ifdefined@c@TF\MT@tr@ospace
2731     {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
2732     {\edef\@tempa{\@tempa,,}}%
2733     \expandafter\MT@tr@set@space\@tempa,%
2734 (pdf-|lua-)
2735 (*letterspace)
2736     % spacing = {<letterspace amount>*,,}
2737     \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
2738     * \fontdimen2\MT@lsfont/1000\relax
2739 (letterspace)

```

Adjust outer kerning (microtype only).

```

2740 (*pdf-|lua-)
2741     \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
2742     \expandafter\MT@tr@set@okern\@tempa,%

```

Disable ligatures (not configurable in `letterspace`).

```

2743     \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
2744 (pdf-|lua-)
2745 (*letterspace)
2746     % no ligatures = {f}
2747     \tagcode\MT@lsfont`f=\m@ne
2748 (letterspace)

```

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

```

2749 (lua-|letterspace)     \MT@if@luaotf@font\relax{%
2750 (debug)\MT@dinfo@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
2751     \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
2752     \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
2753     \let\MT@the@pr@code\MT@the@pr@code@tr
2754 (lua-|letterspace)     }%
2755     \fi

```

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

```

2756     \aftergroup\MT@set@lsfont
2757 (pdf-|lua-)     \let\MT@font\MT@lsfont
2758 (lua-)     \MT@if@luaotf@font\MT@font\relax

```

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

```

\MT@curr@ls 2759     \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
2760     \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.

```

2761 (*pdf-|lua-)

```

```

2762 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
2763 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
2764 \MT@tr@outer@l
2765 </pdf-|lua-|

```

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

```

2766 \ifx\MT@ls@adjust\@empty
2767 <letterspace> % \textls : outer kerning = {*,*}; \textls* : outer kerning = {0,0}
2768 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
2769 \MT@ls@outer@k

```

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

```

2770 <pdf-|lua-|
2771 \else
2772 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
2773 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
2774 \ifdim\MT@outer@kern=\z@\else \MT@ls@outer@k \fi
2775 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
2776 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
2777 </pdf-|lua-|
2778 <letterspace>
2779 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
2780 \MT@afteraftergroup{%
2781 \MT@set@curr@ok
2782 \noexpand\MT@ls@outer@k
2783 }%
2784 </letterspace>
2785 \fi
2786 <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).

```

2787 \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.

```

2788 \MT@afteraftergroup{%
2789 \MT@set@curr@os
2790 \MT@set@curr@ok
2791 \noexpand\MT@tr@outer@r
2792 }%
2793 </pdf-|lua-|
2794 \fi
2795 <pdf-|> }%
2796 }

```

`\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.)

```

2797 \def\MT@afteraftergroup#1{%
2798 <letterspace> \MT@maybe@gobble@with@tikz{%
2799 \MT@ifdefined@n@TF{MT@aftergroup@number\currentgrouplevel}\relax{%
2800 \MT@exp@cs\xdef{MT@aftergroup@number\currentgrouplevel}%
2801 {\MT@exp@cs\MT@glet{MT@aftergroup@number\currentgrouplevel}\noexpand\@undefined#1}%
2802 \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
2803 {MT@aftergroup@number\currentgrouplevel}%
2804 }%
2805 <letterspace> }%
2806 }
2807 </pdf-|lua-|letterspace>

```

```

\MT@ls@fontspec@font      Add the kernfactor feature to a font loaded by fontspec.
2808 (*lua-|letterspace)
2809 \def\MT@ls@fontspec@font{%
2810   \MT@lua{microtype.add_ls[[\MT@letterspace@]]}%
2811 }
2812 (/lua-|letterspace)
2813 (*luafile)
2814 local function add_ls(k)
2815   local f = tex.fontname(font.current())
2816   local spec,size = match(f,'^(.+)( at .+)$')
2817   if not spec then spec = f end
2818   local a,b,c = match(spec,'^([^\:]+):?([^\:]*):?(.*)$')
2819   local ls = "kernfactor=" .. k/1000 .. ' ';
2820   microtype.sprint(a..':')
2821   if (a == "name" or a == "file") then
2822     microtype.sprint(b..'!:'..ls..c)
2823   else
2824     microtype.sprint(ls..b)
2825   end
2826   if size then
2827     microtype.sprint(size)
2828   end
2829 end
2830 microtype.add_ls = add_ls
2831
2832 (/luafile)

\MT@get@tr@opt          Various settings (only for the microtype version).
2833 (*pdf-|lua-)
2834 \def\MT@get@tr@opt{%
2835   \MT@set@listname
2836   \let\MT@tr@factor@\em

\MT@tr@unit@           Different unit (for letterspace and/or (outer)spacing)?
2837   \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name @unit}{%
2838     \MT@let@cn\MT@tr@unit@{MT@tr@cc@MT@tr@cc@name @unit}%
2839     \ifdim\MT@tr@unit@=1em
2840       \let\MT@tr@unit@\undefined
2841     \else
2842       \MT@get@unit\MT@tr@unit@
2843     \fi
2844   }%
2845   \MT@ifdefined@n@T{MT@tr@cc@MT@tr@cc@name}{%
2846     \MT@let@cn\MT@letterspace{MT@tr@cc@MT@tr@cc@name}%
2847     \MT@ifdefined@c@T\MT@tr@unit@{%
2848       \let\@tempb\MT@letterspace
2849       \MT@scale@to@em
2850       \edef\MT@letterspace{\number\@tempcntb}%
2851     }%
2852   }%

\MT@tr@ispace          Adjust interword spacing.
\MT@tr@ospace 2853   \MT@get@tr@opt@{spacing} {ispace}%
2854   \MT@get@tr@opt@{outerspacing}{ospace}%

\MT@tr@okern           Adjust outer kerning.
2855   \MT@get@tr@opt@{outerkerning}{okern}%

\MT@tr@ligatures       Which ligatures should we disable (empty means all, undefined none)?
2856   \MT@get@tr@opt@{noligatures} {ligatures}%
2857 }

\MT@get@tr@opt@
2858 \def\MT@get@tr@opt@#1#2{%

```

```

2859 \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name @#1}%
2860 {\MT@let@nn{MT@tr@#2}{MT@tr@c@MT@tr@c@name @#1}}%
2861 }
2862 </pdf-|lua-|

```

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

```

2863 <pdf-|lua-|letterspace>
2864 <plain>\MT@requires@latex2{
2865 \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@currsiz` 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@currsiz`, fingers crossed. The overhead seems small.

```

2866 \DeclareRobustCommand\lsstyle{%
2867 \not@math@alphabet\lsstyle\textls
2868 \let\glb@currsiz\@empty
2869 <pdf-|lua-| \MT@maybe@gobble@with@tikz{\aftergroup\glb@settings}%
2870 <pdf-|lua-| \def\MT@feat{tr}%
2871 \let\MT@tracking\MT@set@tr@codes
2872 \selectfont
2873 }

```

Now the definitions for the `letterspace` package with plain \TeX .

```

2874 <*plain>
2875 {}
2876 \def\MT@set@lsfont{\MT@lsfont}
2877 \def\lsstyle{%
2878 \begingroup
2879 \escapechar\m@ne
2880 \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
2881 \MT@set@tr@codes
2882 \endgroup
2883 }
2884 \let\textls\@undefined
2885 \let\lslig\@undefined
2886 }
2887 </plain>

```

`\lslig` For Fraktur fonts, some ligatures shouldn't be broken up. This command will temporarily select the base font (making sure to really select the current font) and insert the correct kerning.

```

2888 \DeclareRobustCommand\lslig[1]{%
2889 {\MT@ifdefined@c@TF\MT@curr@ls{%
2890 \escapechar\m@ne
2891 <plain> \MT@requires@latex2{%
2892 \xdef\font@name{\csname\curr@fontshape/\f@size\endcsname}%
2893 <plain> }\relax%
2894 \MT@get@ls@basefont
2895 \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
2896 \kern\MT@outer@kern
2897 \font@name #1%
2898 \kern\MT@outer@kern
2899 }{#1}}%
2900 }

```

`\MT@ls@basefont` pdf \TeX cannot letterspace fonts that already are letterspaced. Therefore, we have 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.

```

2901 \def\MT@get@ls@basefont{%
2902   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2903   \expandafter\ifx\MT@ls@basefont\relax
2904     \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
2905   \else
2906   (debug)\MT@din@fo@n1{1}{... fixing base font}%
2907     \MT@set@ls@basefont
2908   \fi
2909 }

```

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

`\MT@set@tr@zero`

```

2910 \def\MT@set@ls@basefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
2911 \def\MT@set@tr@zero{%
2912   (debug)\MT@din@fo@n1{1}{... zero tracking}%
2913   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2914   \expandafter\ifx\MT@ls@basefont\relax \else
2915   (debug)\MT@din@fo@n1{1}{... fixing base font}%
2916   \aftergroup\MT@set@ls@basefont
2917   \fi
2918 }
2919 (/pdf-|lua-|letterspace)

```

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

```

2920 (*pdf-|lua-)
2921 (pdf-)\MT@requires@pdftex7{
2922   \def\MT@tr@noligatures{%
2923     \ifx\MT@tr@ligatures\@empty
2924       \MT@noligatures@\MT@lsfont\@undefined
2925     \else
2926       \MT@noligatures@\MT@lsfont\MT@tr@ligatures
2927     \fi
2928   }
2929 (*pdf-)
2930 }{
2931   \def\MT@tr@noligatures{%
2932     \MT@warning@n1{%
2933       Disabling selected ligatures is only possible since\MessageBreak
2934       pdftex 1.40.4. Disabling all ligatures instead}%
2935     \MT@gl@et\MT@tr@noligatures\relax
2936   }
2937 }
2938 (/pdf-)

```

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

```
2939 \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.

```

2940 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
2941   (debug)\MT@din@fo@n12{... orig. space: \the\fontdimen2\MT@lsfont,
2942   (debug)   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
2943   (debug)   \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
2944   \let\MT@temp\@empty
2945   \MT@tr@set@space@{#1}{#4}{2}\@empty
2946   \MT@tr@set@space@{#2}{#5}{3}\@pplus
2947   \MT@tr@set@space@{#3}{#6}{4}\@minus
2948   \MT@gl@et@c{MT@outer@space\expandafter\string\font@name}\MT@temp
2949   (debug)\MT@din@fo@n12{... inner space: \the\fontdimen2\MT@lsfont,

```

```

2950 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
2951 <debug>\MT@dinfn12{... outer space: \MT@temp}%
2952 }

```

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

```

2953 \def\MT@tr@set@space@#1#2#3#4{%
2954 \MT@ifempty{#2}{%
2955 \MT@ifempty{#1}\relax{%
2956 \MT@tr@set@space@@{#1}{#3}{1000}%
2957 \fontdimen#3\MT@lsfont=\@tempdima
2958 }%
2959 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
2960 }%
2961 \MT@tr@set@space@@{#2}{#3}{2000}%
2962 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2963 \MT@ifempty{#1}\relax{%
2964 \MT@tr@set@space@@{#1}{#3}{1000}%
2965 \fontdimen#3\MT@lsfont=\@tempdima
2966 }%
2967 }%
2968 }

```

`\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.

```

2969 \def\MT@tr@set@space@@#1#2#3{%
2970 \MT@test@ast#1*\@nil{%
2971 \MT@ifdefined@c@TF\MT@tr@unit@
2972 {\edef\@tempb{#1}\MT@scale@to@em}
2973 {\@tempcntb=#1\relax}%
2974 \@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).

```

2975 \ifnum#2=\tw@
2976 \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
2977 \fi
2978 }%
2979 \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
2980 \@tempdima=\dimexpr\numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
2981 }%
2982 <debug>\MT@dinfn13{... : font dimen #2 (#1): \the\@tempdima}%
2983 }

```

`\MT@tr@outer@` 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).

```

2984 \def\MT@tr@outer@l{%
2985 \ifhmode
2986 \ifdim\lastskip>5sp
2987 \edef\x{\the\lastskip minus 0pt}%
2988 \setbox\z@\hbox{\MT@outer@space=\x}%
2989 \ifdim\wd\z@>\z@
2990 <debug>\MT@dinfn2{[[[ adjusting pre space: \the\MT@outer@space}%
2991 \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

2992 \let\MT@ls@outer@k\relax
2993 \else

```

The ragged2e package sets `\spaceskip` without glue.

```

2994 \ifdim\lastskip=%
2995 \ifnum\spacefactor<2000
2996 \spaceskip
2997 \else

```

```

2998         \ifdim\xspaceskip=\z@
2999         \dimexpr\xspaceskip+\fontdimen7\font@name\relax
3000         \else
3001         \xspaceskip
3002         \fi
3003     \fi
3004 (debug)\MT@dinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
3005         \unskip \hskip\MT@outer@space\relax
3006         \let\MT@ls@outer@k\relax
3007     \fi
3008 \fi
3009 \fi
3010 \fi
3011 }

```

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

```

3012 \def\MT@tr@outer@r{%
3013   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
3014 }

```

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

```

3015 \def\MT@if@outer@next#1{%
3016   \ifx\MT@tr@outer@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
3017 }

```

`\MT@tr@outer@r@`

```

3018 \def\MT@tr@outer@r@{%
3019   \def\MT@temp*{}%

```

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

```

3020   \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).

```

3021     \ifnum\currentgrouptype=10 \else
3022     \def\MT@temp*#1{\ifhmode\hskip\MT@outer@space
3023 (debug)\MT@dinfo2{[[[ adjusting post space (1): \the\MT@outer@space}%
3024     \fi}%
3025     \expandafter\ifcat\expandafter\noexpand\csname MT@tr@outer@next\endcsname\egroup
3026     \ifhmode\unkern\fi\egroup
3027     \MT@set@curr@ok \MT@set@curr@os
3028     \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
3029     \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.

```

3030     \MT@if@outer@next\maybe@ic{%
3031     \MT@set@curr@ok \MT@set@curr@os
3032     \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}%
3033     }%

```

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).

```

3034     \MT@if@outer@next\check@icr{%
3035     \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}%

```



```

3036     }{%
3037     \MT@if@outer@next\@sptoken{%
3038     \def\MT@temp* {\ifhmode\hskip\MT@outer@space
3039 <debug>\MT@dinfor2{]]] adjusting post space (2): \the\MT@outer@space}%
3040     \fi}%
3041     }{%
3042     \MT@if@outer@next~{%
3043     \def\MT@temp*~{\nobreak\hskip\MT@outer@space
3044 <debug>\MT@dinfor2{]]] adjusting post space (3): \the\MT@outer@space}%
3045     }%
3046     }{%
3047     \MT@if@outer@next\ \relax{%
3048     \MT@if@outer@next\space\relax{%
3049     \MT@if@outer@next\@xobeysp\relax{%

```

xspace requires special treatment.

```

3050     \MT@if@outer@next\xspace{%
3051     \def\MT@temp*\xspace{\MT@xspace}%
3052     }{%

```

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

```

3053     \def\MT@temp*{\ifdim\MT@outer@kern=\z@else\MT@ls@outer@k
3054 <debug>\MT@dinfor2{--- adjusting post kern: \the\MT@outer@kern}%
3055     \fi}%
3056     \MT@let@nc{\MT@tr@outer@next}\relax
3057     }}}}]]\fi
3058     \fi\fi
3059     \MT@temp*%
3060 }

```

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

```

\MT@tr@outer@icr@ 3061 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@
3062 \def\MT@tr@outer@icr@{%
3063 \let\@let@token= \MT@tr@outer@next
3064 \maybe@ic@
3065 }

```

`\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.

```

3066 \def\MT@xspace{\futurelet\@let@token\MT@xspace@}
3067 \def\MT@xspace@{\@xspace@firsttrue\@xspace
3068 \ifdim\lastskip>5sp
3069 \unskip \hskip\MT@outer@space
3070 \else
3071 \ifdim\MT@outer@kern=\z@else\MT@ls@outer@k \fi
3072 \fi
3073 }

```

For older pdf_{TEX} versions and Lua_{TEX}, throw an error.

```

3074 }{
3075 \DeclareRobustCommand\lsstyle{%
3076 \MT@error{Letterspacing only works with \MT@engine tex version
3077 <pdf-> 1.40%
3078 <lua-> 0.62%
3079 \MessageBreak or newer}
3080 {Upgrade \MT@engine tex, or try the `soul' package instead.}%
3081 \MT@glet\lsstyle\relax
3082 }
3083 }

```

And for X_YTEX, too.

```

3084 </pdf-|lua->
3085 <*xe->
3086 \DeclareRobustCommand\lsstyle{%

```

```

3087 \MT@error{Letterspacing currently doesn't work with xetex}
3088 {Run pdftex or luatex, or use the `soul' package instead.}%
3089 \MT@gllet\lssstyle\relax
3090 }
3091 (/xe-)

\textls      This command may be used like the other text commands. The starred version
\MT@ls@adjust@ removes kerning on the sides. The optional argument changes the letterspacing
              factor.
3092 (<package|letterspace)
3093 \DeclareRobustCommand\textls{%
3094   \ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%
3095   {\let\MT@ls@adjust@\MT@ls@adjust@relax\MT@textls}%
3096 }

\MT@textls   This is now almost LATEX's \DeclareTextFontCommand, with the difference that we
\MT@letterspace@ adjust the outer spacing and kerning also for \lssstyle, while LATEX's text switches
              don't bother about italic correction.
3097 \newcommand\MT@textls[2] [] {%
3098   \ifmmode
3099     \nfss@text{\MT@ls@set@ls{#1}\lssstyle#2}%
3100   \else
3101     \hmode@bgroup
3102     \MT@ls@set@ls{#1}%
3103     \lssstyle #2%
3104     \expandafter
3105     \egroup
3106   \fi
3107 }

\MT@ls@adjust   Set current letterspacing amount and outer kerning. This has to be done inside the
\MT@ls@adjust@empty same group as the letterspacing command.
\MT@ls@adjust@relax 3108 \def\MT@ls@adjust@empty{\let\MT@ls@adjust@empty}
\MT@ls@set@ls    3109 \def\MT@ls@adjust@relax{\let\MT@ls@adjust@relax}
3110 \def\MT@ls@set@ls#1{%
3111   \MT@ifempty{#1}%
3112   {\let\MT@letterspace@{\undefined}}%
3113   {\KV@sp@def\MT@letterspace@{#1}}%
3114   \edef\MT@letterspace@{\number\MT@letterspace@}%
3115   \MT@ls@too@large\MT@letterspace@}%
3116   \MT@ls@adjust@
3117 }

\MT@ls@too@large   Test whether letterspacing amount is too large.
3118 \def\MT@ls@too@large#1{%
3119   \ifnum#1>\MT@tr@max
3120     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
3121     \edef#1{\number\MT@tr@max}%
3122   \else
3123     \ifnum#1<\MT@tr@min
3124       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
3125       \edef#1{\number\MT@tr@min}%
3126     \fi
3127   \fi
3128 }

\MT@outer@kern   This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern outer kerning.
3129 \newdimen\MT@outer@kern
3130 (/package|letterspace)
3131 (<pdf-|lua-)
3132 \def\MT@tr@set@okern#1,#2,{%
3133   \let\MT@temp@empty

```

```

3134 \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
3135 \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
3136 \MT@gl@et@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
3137 <debug>\MT@din@fo@n12{... outer kerning: (#1,#2)
3138 <debug>          = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%
3139 }

```

\MT@tr@set@okern@

```

3140 \def\MT@tr@set@okern@#1{%
3141 \MT@test@ast#1*\@nil{%
3142 \MT@ifdefined@c@TF\MT@tr@unit@
3143 {\edef\@tempb{#1}\MT@scale@to@em}
3144 {\@tempcntb=#1\relax}%
3145 \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
3146 }{%
3147 \MT@ifempty\@tempa{\let\@tempa\em}\relax
3148 \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
3149 * \fontdimen6\MT@lsfont/2000\relax
3150 }%
3151 \advance\@tempdima -\dimexpr \MT@letterspace@ sp
3152 * \fontdimen6\MT@lsfont/2000\relax
3153 \edef\MT@temp{\MT@temp{\the\@tempdima}}%
3154 }
3155 </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.

```

3156 <*pdf-|lua-|letterspace>
3157 \def\MT@ls@outer@k{%
3158 \ifhmode
3159 \ifdim\lastkern=-3sp \unkern
3160 \ifdim\lastkern=3sp \kern-3sp
3161 \expandafter\expandafter\expandafter\@gobble
3162 \else \unkern
3163 \expandafter\expandafter\expandafter\@firstofone
3164 \fi
3165 \else
3166 \expandafter\@firstofone
3167 \fi
3168 {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
3169 \fi
3170 }
3171 </pdf-|lua-|letterspace>

```

1.2.7 Disabling ligatures

\MT@no@ligatures The possibility to disable ligatures is a new features of pdfT_EX 1.30, and also works with LuaT_EX.

```

3172 <*pdf-|lua-|
3173 <pdf-|>\MT@requires@pdftex5{
3174 \def\MT@no@ligatures{%
3175 \MT@dotrue
3176 \let\@tempa\MT@n1@setname
3177 \MT@map@c@list@n{font,encoding,family,series,shape,size}{%
3178 \MT@ifdefined@n@TF{\MT@checklist@##1}%
3179 {\csname \MT@checklist@##1\endcsname}%
3180 {\MT@checklist@##1}}%
3181 {n1}}%
3182 }%
3183 \ifMT@do
3184 \MT@no@ligatures@MT@font\MT@n1@ligatures
3185 \fi
3186 }

```

```

\MT@noligatures@ This is also used by \MT@set@tr@codes.
3187 <lua->\MT@requires@luatex4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
3188 \def\MT@noligatures@#1#2{%
3189   \MT@ifdefined@c@TF#2{%

```

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

```
3190   \MT@ifdefined@c@TF\tagcode{%
```

No 'inputenc' key.

```

3191     \let\MT@warn@maybe@inputenc\empty
3192     \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
3193     \MT@map@clist@c#2{%
3194       \KV@sp@def\@tempa{#1}\MT@get@slot
3195       \ifnum\MT@char>\m@ne
3196         \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).

```

3197 <lua->   \MT@if@luaotf@font
3198 <lua->   { \MT@lua{microtype.noligatures([[#1]],[[\MT@char]])} }\relax
3199     \fi
3200   }%
3201   \MT@vinfo{... Disabling ligatures for characters: #2}%
3202 }{%
3203   \pdfnoligatures#1%
3204   \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
3205     know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
3206     the font instead}%
3207 }%
3208 }{%
3209   \pdfnoligatures#1%
3210 <lua->   \MT@if@luaotf@font
3211 <lua->   { \MT@lua{microtype.noligatures([[#1]],"_all_")} }\relax
3212   \MT@vinfo{... Disabling all ligatures}%
3213 }%
3214 }
3215 <pdf->}\relax
3216 </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@.

```

3217 <*luafile>
3218 microtype.ligs = microtype.ligs or { }
3219
3220 local function noligatures(fontcs,liga)
3221   local fontcs = match(fontcs,"([^\ ]+)"
3222   microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
3223   table.insert(microtype.ligs[fontcs],liga)
3224 end
3225 microtype.noligatures = noligatures
3226
3227 local function keepligature(c)
3228   local nodedirect = node.direct
3229   local getfield = nodedirect.getfield
3230   local getfont = nodedirect.getfont
3231   local f,ch
3232   if type(c) == "userdata" then -- in older luaotfload versions, c was a node
3233     f = c.font
3234     ch = c.components.char
3235   else -- since 2.6, c is a (direct node) number
3236     f = getfont(c)
3237     ch = getfield(getfield(c,"components"),"char")

```

```

3238 end
3239 -- if ch then -- should always be true
3240 local ligs = microtype.ligs[match(tex.fontidentifier(f), "\\([^\ ]+)")]
3241 if ligs then
3242   for _,lig in pairs(ligs) do
3243     if lig == "_all_" or tonumber(lig) == ch then
3244       return false
3245     end
3246   end
3247 end
3248 return true
3249 -- end
3250 end
3251
3252 if luaotfload and luaotfload.letterspace then
3253   if luaotfload.letterspace.keepligature then
3254     microtype.info("overwriting function `keepligature'")
3255   end
3256   luaotfload.letterspace.keepligature = keepligature
3257 end
3258
3259 /luafile

```

1.2.8 Loading the configuration

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

```

3260 <package|show
3261 <package \def\MT@load@list#1%
3262 <show \def\MTS@load@list#1%
3263   {\edef\@tempa{#1}%
3264     \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
3265     \MT@ifstreq\@tempa\@tempb{%
3266       \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempa' cannot load itself}{}%
3267     }%
3268     \ifx\@tempb\relax
3269 <show :\par\medskip\leavevmode
3270     \else
3271       \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
3272 <show \MTS@printtext{, loading \texttt{\@tempb}}%
3273       \MT@vinfo{... : First loading \@nameuse{MT@abbr@MT@feat} list `@\@tempb'}%
3274       \begingroup
3275         \MT@load@list\@tempb
3276       \endgroup
3277       \edef\MT@curr@list@name{%
3278 <package \@nameuse{MT@abbr@MT@feat} list \noexpand\MessageBreak
3279         `@\@tempb'}%
3280       \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
3281       \expandafter\MT@set@codes\@tempc,\relax,%
3282 <show \vrule width 4cm height .5pt \
3283 <show \MTS@printtext{End of list \texttt{\MT@curr@list@name}}%
3284 <show \par\medskip\leavevmode
3285     }%
3286     \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempb' undefined.\MessageBreak
3287       Cannot load it from list `@\@tempa'}{}%
3288   }%
3289   \fi
3290 }%
3291 }
3292 </package|show

```

`\MT@find@file` Micro-typographic settings may be written into a file `mt-.cfg`.

`\MT@file@list` We must also record whether we've already loaded the file.

```

3293 <package

```

```
3294 \let\MT@file@list\@empty
3295 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```
3296 \MT@in@clist{#1}\MT@file@list
3297 \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.

```
3298 \MT@begin@catcodes
3299 \let\MT@begin@catcodes\relax
3300 \let\MT@end@catcodes\relax
3301 \MT@xadd\MT@file@list{#1,}%
3302 \InputIfFileExists{\MT@cfg@prefix-#1.cfg}{%
3303 \edef\MT@curr@file{\MT@cfg@prefix-#1.cfg}%
3304 \MT@vinfo{... Loading configuration file \MT@curr@file}%
3305 }{%
3306 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
3307 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
3308 \ifMT@inlist@ \else
3309 \InputIfFileExists{\MT@cfg@prefix-\@tempa.cfg}{%
3310 \edef\MT@curr@file{\MT@cfg@prefix-\@tempa.cfg}%
3311 \MT@vinfo{... Loading configuration file \MT@curr@file}%
3312 \MT@xadd\MT@file@list{\@tempa,}%
3313 }{%
3314 \MT@vinfo{... No configuration file \MT@cfg@prefix-#1.cfg}%
3315 }%
3316 \fi
3317 }%
3318 \endgroup
3319 \fi
3320 }
```

`\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.

```
3321 \def\MT@cfg@catcodes{%
3322 \makeatletter
3323 \catcode`\^7%
3324 \catcode`\ 9%
3325 \catcode`\^^I9%
3326 \catcode`\^^M9%
3327 \catcode`\z@
3328 \catcode`\{\@ne
3329 \catcode`\}\@tw@
3330 \catcode`\#6%
3331 \catcode`\%14%
3332 \MT@map@tlist@n
3333 {\!\"$&'\(\)\*\+,\-\.\/\:\;\<=\>?\[\]\_-\|/}%
3334 \makeoother
3335 }
```

`\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.

```
3336 \def\MT@begin@catcodes{%
3337 \begingroup
3338 \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	•	-	•	-	•	-	•	-	•	-	•	-	•	-	-

```
3339 }
```

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

```
3340 \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.

```
3341 \def\MT@get@basefamily#1#2#3#4\nil{%
3342   \ifx\@empty#4%
3343     \def\@tempa{#1#2#3}%
3344   \else
3345     \let\@tempa\@empty
3346     \edef\@tempb{#1#2#3#4}%
3347     \expandafter\MT@get@basefamily@\@tempb\nil
3348   \fi
3349 }
```

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

```
3350 \def\MT@get@basefamily@#1#2\nil{%
3351   \edef\@tempa{\@tempa#1}%
3352   \ifx\@#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
3353   {\MT@in@tlist{#2}\MT@variants
3354   \ifMT@inlist\else\MT@get@basefamily@#2\nil\fi}%
3355 }
```

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

```
\MT@get@listname@ 3356 \def\MT@get@listname#1{%
3357   <debug>\MT@dinfo{n1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
3358   \let\MT@listname\@undefined
3359   \def\@tempb{#1}%
3360   \MT@map@tlist@c\MT@try@order\MT@get@listname@
3361 }
3362 \def\MT@get@listname@#1{%
3363   \expandafter\MT@next@listname#1%
3364   \ifx\MT@listname\@undefined \else
3365     \expandafter\MT@tlist@break
3366   \fi
3367 }
```

```
\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.
```

```
3368 \def\MT@try@order{%
3369   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
3370   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
3371 }
```

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

```

3372 \def\MT@next@listname#1#2#3#4{%
3373   \ifnum#1=\z@\MT@nofamilytrue\fi
3374   \edef\@tempa{\MT@encoding
3375     /\ifnum#1=\@ne \MT@family \fi
3376     /\ifnum#2=\@ne \MT@series \fi
3377     /\ifnum#3=\@ne \MT@shape \fi
3378     /\ifnum#4=\@ne *\fi
3379     \MT@context}%
3380 (debug)\MT@dinfo@n1{1}{trying \@tempa}%
3381   \MT@ifdefined@n@TF{\MT@\@tempb @\@tempa}{%
3382     \MT@next@listname@#4%
3383   }%

```

Also try with an alias family.

```

3384   \ifnum#1=\@ne
3385     \ifx\MT@familyalias\@empty \else
3386       \edef\@tempa{\MT@encoding
3387         /\MT@familyalias
3388         /\ifnum#2=\@ne \MT@series\fi
3389         /\ifnum#3=\@ne \MT@shape\fi
3390         /\ifnum#4=\@ne *\fi
3391         \MT@context}%
3392 (debug)\MT@dinfo@n1{1}{(alias) \@tempa}%
3393   \MT@ifdefined@n@TF{\MT@\@tempb @\@tempa}{%
3394     \MT@next@listname@#4%
3395   }%
3396   \fi
3397 \fi
3398 }%
3399 }

```

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

```

3400 \def\MT@next@listname@#1{%
3401   \ifnum#1=\@ne
3402     \MT@exp@cs\MT@in@rlist{\MT@\@tempb @\@tempa @sizes}%
3403     \ifMT@in@list@
3404       \let\MT@listname\MT@size@name
3405     \fi
3406   \else
3407     \MT@let@cn\MT@listname{\MT@\@tempb @\@tempa}%
3408   \fi
3409 }

```

`\MT@if@list@exists`

```

\MT@context 3410 \def\MT@if@list@exists{%
3411   \MT@let@cn\MT@context{\MT@\MT@feat @context}%
3412   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
3413   \MT@get@listname{\MT@feat @c}%
3414   \MT@ifdefined@c@TF{\MT@listname{%
3415     \MT@edef@n{\MT@\MT@feat @c@name}{\MT@listname}%
3416     \ifMT@nonselected
3417       \MT@vinfo{... Applying non-selected expansion (list `'\MT@listname')}%
3418     \else
3419       \MT@vinfo{... Loading \@nameuse{\MT@abbr@\MT@feat} list `'\MT@listname'}%
3420     \fi
3421     \@firstoftwo
3422   }%

```

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

```

3423   \MT@let@nc{\MT@\MT@feat @c@name}\@empty

```

Don't warn if `selected=false`.


```

3424 \ifMT@nonselected
3425 \MT@vinfo{... Applying non-selected expansion (no list)}%
3426 \else
Tracking doesn't require a list, either.
3427 \MT@ifstreq\MT@feat{tr}\relax{%
3428 \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
3429 for font\MessageBreak`\MT@@font'%
3430 \ifx\MT@context\@empty\else\space(context: `\MT@context')\fi.
3431 Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
3432 }%
3433 \fi
3434 \@secondoftwo
3435 }%
3436 }

```

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

```

\MT@context 3437 \def\MT@get@inh@list{%
3438 \let\MT@context\@empty
3439 \MT@get@listname{\MT@feat @inh}%
3440 \MT@ifdefined@c@TF\MT@listname{%
3441 \MT@edefn{MT@\MT@feat @inh@name}{\MT@listname}%
3442 debug\MT@dinfo@n1{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
3443 debug\MT@dinfo@n1{1}{\MT@listname'}%
3444 \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

3445 \ifx\@tempc\@empty \else
3446 debug\MT@dinfo@n1{1}{parsing inheritance list ...}%

```

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

```

3447 \begingroup
3448 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
3449 \MT@set@inputenc{inh}%
3450 \expandafter\MT@inh@do\@tempc,\relax,%
3451 \MT@gl@et@nc{MT@\MT@feat @inh@\MT@listname}\@empty
3452 \endgroup
3453 \fi
3454 }{%
3455 \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
3456 }%
3457 }

```

1.2.9 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@ 3458 \def\MT@get@slot{%
3459 \escapechar`\
3460 \let\MT@char@m@ne
3461 \MT@norestrue

```

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

```

3462 \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.

```
3463 \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.

```
3464 \expandafter\MT@is@letter\@tempa\relax\relax
3465 \ifnum\MT@char@ < \z@
```

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

If $\langle encoding \rangle \langle command \rangle$ (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.

```
3466 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
3467 \MT@is@symbol
```

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

```
3468 {\expandafter\MT@is@composite\@tempa\relax\relax}%
3469 \ifnum\MT@char@ < \z@
```

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

```
3470 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
3471 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
3472 \fi
3473 \fi

3474 \let\MT@char\MT@char@
3475 \MT@get@slot@
3476 \escapechar\m@ne
3477 }
3478 </package>
```

`\MT@get@slot@`

```
3479 < *pdf- | lua- | xe- >
3480 \def\MT@get@slot@{%
```

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

```
3481 < xe- > \ifnum\XeTeXfonttype\MT@font=\z@
3482 \ifnum\MT@char > \m@ne
```

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

```
3483 < *lua- >
3484 \ifnum\MT@char=47\relax
3485 \ifMT@norest \else
3486 \@tempcnta=\MT@lua{
3487 local glyph = microtype.name_to_slot([[ \expandafter\@gobble\@tempa ]], true)
3488 if glyph then tex.write(glyph)
3489 else tex.write(-1)
3490 end
3491 }\relax
3492 \ifnum\@tempcnta<\z@
3493 \MT@warn@unknown
3494 \let\MT@char\m@ne
3495 \else
3496 \edef\MT@char{\the\@tempcnta}%
```

```

3497 <debug>\MT@info@n1{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
3498     \fi
3499     \fi
3500   \else
3501 </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.

```

3502     \ifMT@noreset \else
3503       \MT@warn@rest
3504 <pdf-|lua->     \let\MT@char\m@ne
3505 <xe->         \let\MT@char\@empty
3506     \fi
3507 </lua->     \fi
3508   \else
3509     \MT@warn@unknown
3510 <xe->     \let\MT@char\@empty
3511   \fi
3512 <*xe->
3513   \else

```

There are more possibilities for Xe_{La}TeX: It may be a Unicode codepoint (prefixed with ‘U’) or a glyph name (prefixed with ‘/’).⁷ We indicate glyph names to \MT@get@charwd by reversing the sign of \MT@char@.

```

3514     \ifnum\MT@char=47\relax
3515       \ifMT@noreset \edef\MT@char{U47}%
3516     \else
3517       \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
3518       \ifnum\@tempcnta=\z@
3519         \MT@warn@unknown
3520         \let\MT@char\@empty
3521       \else
3522         \edef\MT@char{\@tempa\space}%
3523         \edef\MT@char@{-\the\@tempcnta}%
3524 <debug>\MT@info@n1{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
3525     \fi
3526   \fi
3527   \else
3528     \ifnum\MT@char > \m@ne
3529     \ifMT@noreset

```

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.

```

3530     \@tempcnta=\XeTeXcharglyph\MT@char\relax
3531     \ifnum\@tempcnta=\z@
3532       \MT@info@missing@char
3533       \let\MT@char\@empty
3534     \else
3535 <debug>\MT@info@n1{3}{> (glyph number: \the\@tempcnta,
3536 <debug>           glyph name: \XeTeXglyphname\MT@font\@tempcnta)}%
3537       \edef\MT@char{U\MT@char}%
3538     \fi
3539   \else
3540     \MT@warn@rest
3541     \let\MT@char\@empty
3542   \fi
3543   \else
3544     \MT@warn@unknown
3545     \let\MT@char\@empty
3546   \fi

```

⁷ This doesn’t seem to be documented anywhere, but it has been announced here: <https://tug.org/pipermail/xetex/2010-May/016531.html>

```

3547   \fi
3548   \fi
3549 </xe-
3550 }
3551 </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.

```

3552 <luafile>
3553 if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
3554   local slot_of_name = luaotfload.aux.slot_of_name
3555   microtype.name_to_slot = function(name, unsafe)
3556     local n = slot_of_name(font.current(), name, unsafe)
3557     if not n then return -1 end
3558     if n > 1114111 then return -1 end
3559     return math.tointeger(n)
3560   end
3561 else
3562   -- we dig into internal structure (should be avoided)
3563   local function name_to_slot(name, unsafe)
3564     if fonts then
3565       local unicodes
3566       if fonts.ids then -- legacy luaotfload
3567         local tfmdata = fonts.ids[font.current()]
3568         if not tfmdata then return end
3569         unicodes = tfmdata.shared.otfdata.luatex.unicodes
3570       else -- new location
3571         local tfmdata = fonts.hashes.identifiers[font.current()]
3572         if not tfmdata then return end
3573         unicodes = tfmdata.resources.unicodes
3574       end
3575       local unicode = unicodes[name]
3576       if unicode then -- does the 'or' branch actually exist?
3577         return type(unicode) == "number" and unicode or unicode[1]
3578       end
3579     end
3580   end
3581   microtype.name_to_slot = name_to_slot
3582 end
3583
3584 </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 3585 <pdf-|lua-|xe-
3586 \def\MT@max@char
3587 <pdf- > {127 }
3588 <lua-|xe- > {1114111 }
3589 \def\MT@max@slot
3590 <pdf- > {255 }
3591 <lua-|xe- > {1114111 }
3592 </pdf-|lua-|xe-

\ifMT@noest    Test whether all of the string has been used up.
3593 <package>
3594 \newif\ifMT@noest
3595 \def\MT@is@letter#1#2\relax{%
3596   \ifcat a\noexpand#1\relax
3597     \edef\MT@char@{\number`#1}%

```

```

3598 \ifx\#2\%
3599 (debug)\MT@info@n1{3}> `the\MT@toks' is a letter (\MT@char@)%
3600 \else
3601 \MT@noestfalse
3602 \fi
3603 \else
3604 \ifcat !\noexpand#1\relax
3605 \edef\MT@char@\number`#1%
3606 (debug)\MT@info@n1{3}> `the\MT@toks' is a character (\MT@char@)%
3607 \ifx\#2\%
3608 \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
3609 \else
3610 \MT@noestfalse
3611 \expandafter\MT@is@number#1#2\relax\relax
3612 \fi
3613 \fi
3614 \fi
3615 }

```

`\MT@is@number` Numbers may be specified as a three-digit decimal number (029), as a hexadecimal number (prefixed with " : "1D) or as an octal number (prefixed with ' : '35). They must consist of at least three characters (including the prefix), that is, "F is not permitted.

```

3616 \def\MT@is@number#1#2#3\relax{%
3617 \ifx\relax#3\relax \else
3618 \ifx\relax#2\relax \else
3619 \MT@noesttrue
3620 \if#1"\relax
3621 \def\x{\uppercase{\edef\MT@char@\number#1#2#3}}\x
3622 (debug)\MT@info@n1{3}> ... a hexadecimal number: \MT@char@)%
3623 \else
3624 \if#1'\relax
3625 \def\MT@char@\number#1#2#3%
3626 (debug)\MT@info@n1{3}> ... an octal number: \MT@char@)%
3627 \else
3628 \MT@ifint{#1#2#3}{%
3629 \def\MT@char@\number#1#2#3%
3630 (debug)\MT@info@n1{3}> ... a decimal number: \MT@char@)%
3631 } \MT@noestfalse
3632 \fi
3633 \fi
3634 \ifnum\MT@char@ > \MT@max@slot
3635 \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
3636 \let\MT@char@\m@ne
3637 \fi
3638 \fi
3639 \fi
3640 }

```

`\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.

```

3641 \def\MT@is@active#1#2\@nil{%
3642 \ifnum\catcode`#1 = \active
3643 \begingroup

```

```

3644 \set@display@protect
3645 \let\IeC\@firstofone
3646 \let@inpenc@undefined@MT@undefined@char

```

Unicode handling has changed again with L^AT_EX 2019/10/01.

```

3647 \let\UTF@two@octets@noexpand@empty
3648 \let\UTF@three@octets@noexpand@empty
3649 \let\UTF@four@octets@noexpand@empty

```

We refrain from checking whether there is a sufficient number of octets.

```

3650 \def\UTFviii@defined##1{\ifx ##1\relax
3651 \MT@undefined@char{utf8}\else\expandafter ##1\fi}%

```

For ucs (utf8x). Let's call it experimental ...

```

3652 \MT@ifdefined@c@T\PrerenderUnicode
3653 {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
3654 \MT@is@active@hook{#1}%

```

The \expandafter hocus-pocus should please newunicodechar.

```

3655 \edef\x{\endgroup
3656 \def\noexpand\@tempa{\expandafter\expandafter\expandafter\@empty\@tempa}%

```

Append what we think the translation is to the token register we use for the log.

```

3657 \MT@toks={\the\MT@toks\space(=
3658 \expandafter\expandafter\expandafter\@empty\@tempa)}%
3659 }%
3660 \x
3661 \fi
3662 }

```

\MT@is@active@hook Test for these packages only once (requires etoolbox).

```

3663 \let\MT@is@active@hook@gobble
3664 ^^Q\@gobble
3665 {\catcode`\#=12
3666 \MT@addto@setup}%

```

If a char has been made active by listings's \lstMakeShortInline, we need to retrieve the original meaning, or else make sure that we're seeing a non-active char.

```

3667 \MT@with@package@T{listings}{%
3668 \apptocmd\MT@is@active@hook{%
3669 \MT@ifdefined@n@T{lst@ShortInlineOldCatcode\string#1}{%
3670 \catcode`#1=\csname lst@ShortInlineOldCatcode\string#1\endcsname\relax
3671 \ifnum\catcode`#1=\active
3672 \begingroup
3673 \catcode`\-\active \lccode`\-`#1%
3674 \lowercase{\endgroup
3675 \MT@let@cn-{\lst@ShortInlineOldMeaning\string#1}}%
3676 \else
3677 \def\@tempa{#1}%
3678 \fi
3679 }%
3680 }{}{}%
3681 }%

```

Same for \MakeShortVerb of doc/shortvrb (and implicitly memoir).

```

3682 \MT@if@false
3683 \MT@with@package@T{doc}\MT@if@true
3684 \MT@with@package@T{shortvrb}\MT@if@true
3685 \ifMT@if\expandafter\@firstofone\else\expandafter\@gobble\fi{%
3686 \apptocmd\MT@is@active@hook{%
3687 \MT@ifdefined@n@T{cc\string#1}{%
3688 \catcode`#1=\csname cc\string#1\endcsname\relax
3689 \ifnum\catcode`#1=\active
3690 \begingroup
3691 \catcode`\-\active \lccode`\-`#1%

```

```

3692         \lowercase{\endgroup
3693         \MT@let@cn-{\ac\string#1}}%
3694     \else
3695         \def\@tempa{#1}%
3696     \fi
3697 }%
3698 }{}{}%
3699 }%
3700 }}

```

`\MT@undefined@char` For characters not defined in the current input encoding.

```

3701 \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.

```

3702 \def\MT@is@symbol{%
3703   \expandafter\def\expandafter\MT@char\expandafter
3704     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%

```

Since recently, some glyphs are defined optionally in L^AT_EX by checking if the glyph actually exists in the font (e.g., `\textasteriskcentered`).

```

3705   \expandafter\expandafter\expandafter
3706     \MT@is@opt@char\MT@char\iffontchar\char\else\fi\relax
3707   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
3708     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
3709   \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.

```

3710   \expandafter\expandafter\expandafter\MT@is@tlig\MT@char\relax\relax
3711   \ifnum\MT@char@ < \z@

```

Finally, if it hasn't been defined by `\DeclareTextSymbol`, it could be a letter (e.g., `\i`, when using `frenchpro`).

```

3712   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
3713   \fi
3714   \fi
3715 }

```

`\MT@is@opt@char` This seems adventurous, but we're only redefining the text command within the scope of our setup.

```

3716 \def\MT@is@opt@char#1\iffontchar#2\char#3\else#4\fi\relax{%
3717   \MT@ifempty{#1}{%
3718     \iffontchar#2%
3719     \MT@exp@cs\chardef{\MT@encoding\MT@detokenize@c\@tempa}=#3\relax
3720   \fi
3721 } \relax
3722 }

```

`\MT@is@char` A helper macro that inspects the `\meaning` of its argument.

```

\MT@charstring 3723 \begingroup
3724   \catcode`\/= \z@
3725   /MT@map@tlist@n{/CHARLEX}/@makeoether
3726   /lowercase{%
3727     /def/x{/endgroup
3728     /def/MT@charstring{\CHAR"%
3729     /def/MT@is@char##1\CHAR"##2##3##4/relax{%
3730     /ifx/relax##4/relax

```

```

3731         /ifMT@xunicode
3732         /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
3733         /relax/relax/relax/relax/relax
3734     /fi
3735 /else
3736 /ifx/relax##1/relax
3737 /if##3\relax
3738     /edef/MT@char@{/number"##2}%
3739     /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
3740 /else
3741     /edef/MT@char@{/number"##2##3}%
3742     /MT@ifstreq/MT@charstring{##4}/relax
3743     {/MT@is@xchar##2##3|##4\CHAR"/relax}%
3744 /fi
3745 (debug) /MT@dinfo@n1{3}{> `~/the/MT@toks' is a \char (/MT@char@)}%
3746 /fi
3747 /fi
3748 }%

```

`\MT@is@xchar` With fontspec's TU encoding, glyph numbers may be up to four digits.

```

3749     /def/MT@is@xchar##1|##2\CHAR"##3##4/relax{%
3750     /MT@ifstreq/MT@charstring{##3##4}%
3751     /edef/MT@char@{/number"##1##2}/MT@norestfalse
3752 }%

```

`\MT@charxstring` For xunicode, which doesn't `\countdef`, but rather `\def`s the chars.

```

\MT@strip@prefix 3753     /def/MT@charxstring{\CHAR "%
\MT@is@charx     3754     /def/MT@strip@prefix##1>##2/relax{##2}%
                 3755     /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
                 /ifx/relax##1/relax
                 /ifx/relax##6/relax/else
                 /edef/MT@char@{/number"##2##3##4##5}%
                 /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@norestfalse
3760 (debug) /MT@dinfo@n1{3}{> `~/the/MT@toks' is a xunicode \char (/MT@char@)}%
                 /fi
                 /fi
                 }%
                 }%
3764     }%
3765 }
3766 /x

```

`\MT@is@tlig` This might have to change again with the next L^AT_EX release, ... or so I feared, but it still seems to be fine.

```

3767 \def\MT@is@tlig#1#2\relax{%
3768   \ifx\remove@tlig#1
3769 (debug) \MT@dinfo@n1{3}{> `~/the/MT@toks' (removing remove@tlig)}%
3770   \MT@remove@tlig
3771   \fi
3772 }

```

`\MT@remove@tlig` We remove the `\remove@tlig` command and only pass on the number.

```

3773 \def\MT@remove@tlig{%
3774   \expandafter\MT@exp@two@c\expandafter\MT@is@number
3775   \expandafter\@secondoftwo\MT@char\relax\relax
3776 }

```

`\MT@is@composite` Here, we are dealing with accented characters, specified as two tokens.

```

3777 \def\MT@is@composite#1#2\relax{%
3778   \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`.

```
3779 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
3780 \string\csname\MT@encoding\endcsname
3781 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
```

In 2017, L^AT_EX introduced a new way of declaring accented Unicode commands (`\DeclareUnicodeComposite`), which we take care of here (`\UnicodeEncodingName` has been introduced at the same time):

```
3782 \ifx\UnicodeEncodingName\undefined\else
3783 \expandafter\expandafter\expandafter
3784 \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
3785 \fi
3786 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

Again, `xunicode`.

```
3787 \ifnum\MT@char@ < \z@
3788 \ifMT@xunicode
3789 \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
3790 \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
3791 \MT@char\MT@charxstring\relax\relax\relax\relax\relax
3792 \fi
3793 \fi
3794 \fi
3795 }
```

`\MT@is@uni@comp` Helper for `\DeclareUnicodeComposite`.

```
3796 \def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
3797 \ifx\#1\edef\MT@char{\iffontchar#2\fi}\fi
3798 }
```

[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@list@name 3799 \def\MT@set@list@name{%
3800 \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list\noexpand\MessageBreak
3801 \~\@nameuse{MT@MT@feat @c@name}'}%
3802 }
```

`\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.

```
3803 \def\MT@warn@ascii{%
3804 \MT@warning@n1{Character `the\MT@toks' (= \MT@char@)
```

```

3805     is outside of ASCII range.\MessageBreak
3806     You must load the \inputenc package before using\MessageBreak
3807     8-bit characters in \MT@curr@list@name}%
3808 }

\MT@warn@number@too@large    Number too large.
3809 \def\MT@warn@number@too@large#1{%
3810   \MT@warning@nl{%
3811     Number #1 in encoding \MT@encoding' too large!\MessageBreak
3812     Ignoring it in \MT@curr@list@name}%
3813 }

\MT@warn@rest    Not all of the string has been parsed.
3814 \def\MT@warn@rest{%
3815   \MT@warning@nl{%
3816     Unknown slot number of character\MessageBreak\the\MT@toks'
3817     \MT@warn@maybe@inputenc\MessageBreak
3818     in font encoding \MT@encoding'.\MessageBreak
3819     Make sure it's a single character\MessageBreak
3820     (or a number) in \MT@curr@list@name}%
3821 }

\MT@warn@unknown    No idea what went wrong.
3822 \def\MT@warn@unknown{%
3823   \MT@warning@nl{%
3824     Unknown slot number of character\MessageBreak\the\MT@toks'
3825     \MT@warn@maybe@inputenc\MessageBreak
3826     in font encoding \MT@encoding' in \MT@curr@list@name}%
3827 }

\MT@warn@maybe@inputenc    In case an input encoding had been requested.
3828 \def\MT@warn@maybe@inputenc{%
3829   \MT@ifdefined@n@T
3830   {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
3831   { (input encoding \@nameuse
3832     {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
3833 }

```

1.2.10 Hook into L^AT_EX's font selection

We append `\MT@setupfont` to `\pickup@font`, which is called by L^AT_EX 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 `pdfcpot` 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`, `pi font` (loaded by `hyperref`), `tifa`, 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 3834 \let\MT@font@list\@empty
3835 \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`.

```
3836 </package>
3837 <*package|letterspace>
3838 <plain>\MT@requires@latex2{
3839 \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.

```
3840 <package> \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
3841 <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`.

```
3842 \ifpackage@loaded{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.

```
3843 \ifpackage@loaded{xeCJK}{\@firstofone}{%
3844 \ifpackage@later{CJK}{2006/10/17}% 4.7.0
3845 {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
3846 {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
3847 \g@addto@macro\MT@orig@pickupfont
3848 {\@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).

```
3849 \ifpackage@loaded{CJKutf8}%
3850 {\@ifpackage@later{CJKutf8}{2008/05/22}% 4.8.0
3851 {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}}%
```

```

3852     {\@firstoftwo}}%
3853     {\@firstoftwo}%
3854     {\g@addto@macro\MT@orig@pickupfont{%
3855     {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
3856     \define@newfont\else\xdef\font@name{%
3857     \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
3858     {\g@addto@macro\MT@orig@pickupfont{%
3859     {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
3860     \define@newfont\def\CJK@temp{v}%
3861     \ifx\CJK@temp\CJK@plane
3862     \expandafter\ifx\csname CJK@cm@p@\f@family\CJK@plane\endcsname\relax
3863     \else\csname CJK@cm@p@\f@family\CJK@plane\endcsname\fi
3864     \else \CJK@addcm@p\CJK@plane \fi
3865     \else\xdef\font@name{%
3866     \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
3867     \@gobble
3868     }%
3869   }{\@firstofone}%

```

This is the normal L^AT_EX definition.

```

3870   {\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.

```

3871   \ifx\pickup@font\MT@orig@pickupfont \else
3872     \MT@warning@nl{%
3873     Command \string\pickup@font\space is not defined as expected.%
3874     \MessageBreak Patching it anyway. Some things may break%
3875   }{*package}
3876     .\MessageBreak Double-check whether micro-typography is indeed%
3877     \MessageBreak applied to the document.%
3878     \MessageBreak (Hint: Turn on `verbose' mode)%
3879   }{/package}
3880   }%
3881   \fi

```

`\pickup@font` Then we append our stuff. Everything is done inside a group.

```

3882   \g@addto@macro\pickup@font{\begingroup}%

```

If the `trace` package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.

```

3883   \MT@with@package{T{trace}}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
3884   \g@addto@macro\pickup@font{%

```

If we're inside an `\edef` (or `\write ...`), we don't want to execute our code. This will still leave `'\begingroup \let \relax \relax \endgroup'` in the input stream, which is not nothing but should be harmless enough. `\pickup@font` should never be executed in these contexts anyway, but obviously this may, under rare circumstances, still happen (e.g., with `hyperref`).⁸

```

3885 {package}   \MT@if@expanding@F{%
3886           \escapechar\m@ne
3887 {*package}
3888 {debug}     \global\MT@inannottrue
3889 {debug}     \MT@glet\MT@pdf@annot\@empty
3890 {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.

⁸ Cf. <https://tex.stackexchange.com/q/687763/7674>

```

3891     \MT@let@cn\MT@font{MT@subst@expandafter\string\font@name}%
3892     \ifx\MT@font\relax
3893         \let\MT@font\font@name
3894     \else
3895         \ifx\MT@font\font@name \else
3896 <debug> \MT@addto@annot{= substituted with \MT@font}%
3897         \MT@register@subst@font
3898         \fi
3899     \fi
3900     \MT@setupfont}%
3901 </package>
3902 <letterspace> \MT@tracking
3903 \endgroup
3904 }%
3905 <*package>

```

`\MT@pickupfont` Remember the patched command, because we may have to disable ourselves in certain situations.

`\MT@MT@pickupfont`

```

\MT@ltx@pickupfont 3906 \let\MT@pickupfont\pickup@font
3907 \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
3908 \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.

```

3909 \g@addto@macro\do@subst@correction
3910 {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
3911 \MT@glet@nc{MT@subst@expandafter\string\font@name}\MT@font}%

```

`\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, L^AT_EX takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

`\MT@orig@add@accent`

```

3912 \let\MT@orig@add@accent\add@accent
3913 \def\add@accent#1#2{%
3914     \MT@ltx@pickupfont
3915     \MT@orig@add@accent{#1}{#2}%
3916     \MT@MT@pickupfont
3917 }%
3918 </package>
3919 }
3920 <plain>\relax
3921 </package|letterspace>
3922 <*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.

```

3923 \def\MT@check@font{\MT@exp@one@n\MT@in@cl@st\MT@font\MT@font@list}

```

`\MT@register@font` Register the current font.

```

3924 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}

```

`\MT@register@subst@font` Register the substituted font (only if isn't registered already). Additionally, we have to remove the substitute font from the list of fonts, so that we set it up again.

```

3925 \def\MT@register@subst@font{%
3926     \MT@exp@one@n\MT@in@cl@st\font@name\MT@font@list
3927     \if\MT@in@list@
3928         \xdef\MT@font@list{\MT@font@list\font@name,}%

```

```

3929 \expandafter\MT@rem@from@clist\MT@font\MT@font@list
3930 \fi
3931 }
3932 </package>

```

1.2.11 Context-sensitive setup

Here are the variants for context-sensitive setup.

`\MT@active@features` The activated features are stored in a command. We always allow contexts for tracking, because `\textls` may be used without activating the feature.

```

3933 <*pdf-|lua-|xe-
3934 pdf-)\MT@requires@pdftex6
3935 lua-)\MT@requires@luatex3
3936 pdf-|lua-|xe- \def\MT@active@features{,tr}}{%
3937 \let\MT@active@features@empty
3938 pdf-|lua-|xe- }
3939 </pdf-|lua-|xe-

```

`\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.

```

3940 <*package>
3941 \def\MT@check@font@cx{%
3942 \MT@if@true
3943 \MT@map@clist@c\MT@active@features{%
3944 \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
3945 \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
3946 \ifMT@inlist@
3947 \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
3948 \else
3949 \MT@if@false
3950 \fi
3951 }%
3952 \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
3953 }

```

`\MT@register@subst@font@cx` Add the substituted font to each feature list and possibly remove substitute font.

```

3954 \def\MT@register@subst@font@cx{%
3955 \MT@map@clist@c\MT@active@features{%
3956 \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
3957 \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
3958 \ifMT@inlist@ \else
3959 \MT@exp@cs\MT@xadd
3960 {MT@##1\csname MT@##1@context\endcsname font@list}%
3961 {\font@name,}%
3962 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font
3963 \csname MT@##1\csname MT@##1@context\endcsname font@list\endcsname
3964 \fi
3965 }%
3966 }

```

`\MT@register@font@cx` For each feature, add the current font to the list, unless we didn't set it up.

```

3967 \def\MT@register@font@cx{%
3968 \MT@map@clist@c\MT@active@features{%
3969 \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
3970 \MT@exp@cs\MT@xadd
3971 {MT@##1\csname MT@##1@context\endcsname font@list}%
3972 {\MT@font,}%
3973 \def\@tempa{##1}%
3974 \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
3975 \fi

```

```

3976 }%
3977 }

\MT@maybe@rem@from@list Recurse through all context font lists of the document and remove the font, unless
                           it's the current context.
3978 \def\MT@maybe@rem@from@list#1{%
3979   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
3980     \expandafter\MT@expone@n\expandafter\MT@rem@from@clist\expandafter
3981     \MT@font \csname MT@\@tempa @#1font@list\endcsname
3982   }%
3983 }

\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.
3984 \def\microtypecontext{\MT@begin@catcodes\MT@microtypecontext}
3985 \def\MT@microtypecontext#1{\MT@end@catcodes\MT@addto@setup{\microtypecontext{#1}}
3986 \MT@addto@setup{%
3987   \DeclareRobustCommand\microtypecontext{%
3988     \MT@begin@catcodes
3989     \MT@microtypecontext
3990   }%
3991   \def\MT@microtypecontext#1{%
3992     \MT@end@catcodes
3993     \MT@setup@contexts
3994     \let\MT@reset@context\relax

                       We need to ensure that math fonts are set up anew.
3995     \MT@glet@glb@currsizel@empty
3996     \setkeys{MTC}{#1}%
3997     \selectfont
3998     \MT@reset@context
3999   }%
4000 }

\textmicrotypecontext This is just a wrapper around \microtypecontext.
\MT@textmicrotypecontext 4001 \DeclareRobustCommand\textmicrotypecontext{\MT@begin@catcodes\MT@textmicrotypecontext}
\MT@text@microtypecontext 4002 \def\MT@textmicrotypecontext#1{\MT@end@catcodes\MT@text@microtypecontext{#1}}
4003 \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.
4004 \def\MT@reset@context@{%
4005   \MT@vinfo{<<< Resetting contexts\on@line
4006   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
4007   <debug> \MT@tr@context/\MT@kn@context/\MT@sp@context
4008   }%
4009   \selectfont
4010 }

\MT@setup@contexts The first time \microtypecontext is called, we initialise the context lists and
                    redefine the commands used in \pickup@font.
4011 \def\MT@setup@contexts{%
4012   \MT@map@clist@c\MT@active@features
4013   {\MT@glet@nc{MT@##1@font@list}\MT@font@list}%
4014   \MT@glet\MT@check@font\MT@check@font@cx
4015   \MT@glet\MT@register@font\MT@register@font@cx
4016   \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
4017   \MT@glet\MT@setup@contexts\relax
4018 }

                    Define context keys.

```

```

4019 \MT@map@clist@c\MT@features@long{%
4020   \define@key{MTC}{#1} [] {%
4021     \edef@tempb{\@nameuse{MT@rbba@#1}}%
4022     \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
4023     \ifMT@in@list@

```

Using an empty context is only asking for trouble, therefore we choose the ‘@’ instead (hoping for the L^AT_EX users’ natural awe of this character).

```

4024   \MT@ifempty{#1}{\def\MT@val{0}}{\def\MT@val{#1}}%
4025   \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
4026 (debug)\MT@dinfo{1}{>>> no change of #1 context: `~\MT@val'}%
4027   \else
4028     \MT@vinfo{>>> Changing #1 context to `~\MT@val'\MessageBreak\on@line
4029 (debug)     \space(previous: `~\@nameuse{MT@\@tempb @context}')}%
4030     }%
4031   \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

The next time we see the font, we have to reset *all* factors.

```

4032   \MT@gl@et@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%

```

We must also keep track of all contexts in the document.

```

4033   \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
4034   \MT@val \csname MT@\@tempb @doc@contexts\endcsname
4035   \ifMT@in@list@ \else
4036     \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}%
4037 (debug) \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
4038   \fi
4039   \MT@edef@n{MT@\@tempb @context}{\MT@val}%
4040   \fi
4041   \fi
4042 }%
4043 }

```

We also allow the activate shortcut.

```

4044 \define@key{MTC}{activate} [] {%
4045   \setkeys{MTC}{protrusion=#1}}%
4046 \setkeys{MTC}{expansion=#1}}%
4047 }

```

`\MT@pr@context` Initialise the contexts.

```

\MT@ex@context 4048 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl} {%

```

```

\MT@tr@context 4049 \MT@def@n{MT@#1@context}{@}%

```

```

\MT@sp@context 4050 \MT@def@n{MT@#1@doc@contexts}{\@}%

```

```

\MT@kn@context 4051 }

```

```

\MT@kn@context 4052 \let\MT@extra@context\@empty

```

```

\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*

```

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.

```

4053 \def\DeclareMicrotypeSet{%
4054   \MT@begin@catcodes
4055   \ifstar
4056     \MT@DeclareSetAndUseIt
4057     \MT@DeclareSet
4058 }

```


`\MT@DeclareSet`

```

4059 \newcommand\MT@DeclareSet[3] [] {%
4060   \MT@ifempty{#1}{%
4061     \MT@map@clist@{\MT@features{\begingroup\MT@declare@sets{##1}{#2}{#3}\endgroup}%
4062   }{%
4063     \MT@map@clist@{#1}{\begingroup
4064       \MT@ifempty{##1}\relax{%
4065         \MT@is@feature{##1}{set declaration `#2'}{%
4066           \MT@exp@one@{\MT@declare@sets
4067             {\csname MT@rbba@##1\endcsname}{#2}{#3}%
4068           }%
4069         }%
4070       \endgroup}%
4071     }%
4072   \MT@end@catcodes
4073 }

```

`\MT@DeclareSetAndUseIt`

```

4074 \newcommand\MT@DeclareSetAndUseIt[3] [] {%
4075   \MT@DeclareSet[#1]{#2}{#3}%
4076   \UseMicrotypeSet[#1]{#2}%
4077 }

```

`\MT@curr@set@name` We need to remember the name of the set currently being declared.

```
4078 \let\MT@curr@set@name\empty
```

`\MT@declare@sets` Define the current set name and parse the keys.

```

4079 \def\MT@declare@sets#1#2#3{%
4080   \def\MT@curr@set@name{#2}%
4081   \MT@ifdefined@n@T{MT@#1@set@{\MT@curr@set@name}}{%
4082     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set ` \MT@curr@set@name'}%
4083     \MT@map@clist@{font,encoding,family,series,shape,size}{%
4084       \MT@gl@et@nc{MT@#1@list@##1@{\MT@curr@set@name}}\@undefined
4085     }%
4086   }%
4087   \MT@gl@et@nc{MT@#1@set@{\MT@curr@set@name}}\@empty
4088   (debug)\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set ` \MT@curr@set@name'}%
4089   \setkeys{MT@#1@set}{#3}%
4090 }

```

`\MT@define@set@key@` `<#1> = font axis, <#2> = feature.`

```

4091 \def\MT@define@set@key@#1#2{%
4092   \define@key{MT@#2@set}{#1} [] {%
4093     \MT@gl@et@nc{MT@#2@list@#1@{\MT@curr@set@name}}\@empty
4094     \MT@map@clist@{##1}{%
4095       \KV@sp@def\MT@val{###1}%
4096       \MT@get@highlevel{#1}%

```

We do not add the expanded value to the list ...

```

4097   \MT@exp@two@{\g@addto@macro
4098     {\csname MT@#2@list@#1@{\MT@curr@set@name}\expandafter\endcsname}%
4099     {\MT@val,}%
4100   }%

```

... but keep in mind that the list has to be expanded at the end of the preamble.

```

4101   \expandafter\g@addto@macro\expandafter\MT@font@sets
4102   \csname MT@#2@list@#1@{\MT@curr@set@name}\endcsname
4103   (debug)\MT@dinfnl{1}{-- #1: \@nameuse{MT@#2@list@#1@{\MT@curr@set@name}}}%
4104   }%
4105 }

```

`\MT@get@highlevel` Saying, for instance, ‘family=rm*’ or ‘shape=bf*’ will expand to `\rmdefault` resp. `\bfdefault`.

```
4106 \def\MT@get@highlevel#1{%
```

```

4107 \expandafter\MT@test@ast\MT@val*\@nil\relax{%
      And ‘family = *’ will become \familydefault.
4108 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
      Test whether the command is actually defined.
4109 \MT@ifdefined@n@TF{\@tempa default}%
4110 {\edef\MT@val{\MT@exp@cs\noexpand{\@tempa default}}}%
4111 {\MT@warning{\@backslashchar\@tempa default' is not a defined command.\MessageBreak
4112             Ignoring `#1 = {\@tempa*}' in font set\MessageBreak`MT@curr@set@name'}%
4113 \let\MT@val\@empty}%

```

In contrast to earlier versions, these values will not be expanded immediately, but at the end of the preamble.

```

4114 }%
4115 }

```

`\MT@test@ast` It the last character is an asterisk, execute the second argument, otherwise the first one.

```

4116 \def\MT@test@ast#1*#2\@nil{%
4117 \def\@tempa{#1}%
4118 \MT@ifempty{#2}%
4119 }

```

`\MT@font@sets` Fully expand the font specification and fix catcodes for all font sets. Also remove
`\MT@fix@font@set` fontspec’s counters.

```

4120 \let\MT@font@sets\@empty
4121 \def\MT@fix@font@set#1{%
4122 \MT@ifdefined@c@T{#1}{%
4123 \xdef#1{#1}%
4124 \ifMT@fontspec
4125 \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
4126 \fi
4127 \global\@onelevel@sanitize#1%
4128 }%
4129 }

```

`\MT@define@set@key@size` size requires special treatment.

```

4130 \def\MT@define@set@key@size#1{%
4131 \define@key{MT@#1@set}{size}[]{}%
4132 \MT@map@clist@n{##1}{%
4133 \def\MT@val{###1}%
4134 \expandafter\MT@get@range\MT@val--\@nil
4135 \ifx\MT@val\relax \else
4136 \MT@exp@cs\MT@xadd
4137 {MT@#1list@size@MT@curr@set@name}%
4138 {{{\MT@lower}{\MT@upper}\relax}}%
4139 \fi
4140 }%
4141 <debug>\MT@dinfo@n1{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
4142 }%
4143 }

```

Font sizes may also be specified as ranges. This has been requested by Andreas Böhmann, 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` Ranges will be stored as triplets of `{\lower bound}{\upper bound}{\list name}`.

`\MT@upper` For simple sizes, the upper boundary is `-1`.

```

\MT@lower 4144 \def\MT@get@range#1-#2-#3\@nil{%
4145 \MT@ifempty{#1}{%

```

```

4146 \MT@ifempty{#2}{%
4147 \let\MT@val\relax
4148 }{%
4149 \def\MT@lower{0}%
4150 \def\MT@val{#2}%
4151 \MT@get@size
4152 \edef\MT@upper{\MT@val}%
4153 }%
4154 }{%
4155 \def\MT@val{#1}%
4156 \MT@get@size
4157 \ifx\MT@val\relax \else
4158 \edef\MT@lower{\MT@val}%
4159 \MT@ifempty{#2}{%
4160 \MT@ifempty{#3}%
4161 {\def\MT@upper{-1}}%

```

2048 pt is TeX's maximum font size.

```

4162 {\def\MT@upper{2048}}%
4163 }{%
4164 \def\MT@val{#2}%
4165 \MT@get@size
4166 \ifx\MT@val\relax \else
4167 \MT@ifdim\MT@lower>\MT@val{%
4168 \MT@error{%
4169 Invalid size range (\MT@lower\space > \MT@val) in font set
4170 ~\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
4171 \edef\MT@upper{\MT@lower}%
4172 \edef\MT@lower{\MT@val}%
4173 }{%
4174 \edef\MT@upper{\MT@val}%
4175 }%
4176 \MT@ifdim\MT@lower=\MT@upper
4177 {\def\MT@upper{-1}}%
4178 \relax
4179 \fi
4180 }%
4181 \fi
4182 }%
4183 }

```

`\MT@get@size` Translate a size selection command and normalise it.

```
4184 \def\MT@get@size{%
```

A single star would mean `\sizedefault`, which doesn't exist, so we define it to be `\normalsize`.

```

4185 \if*\MT@val\relax
4186 \def\@tempa{\normalsize}%
4187 \else
4188 \MT@let@cn\@tempa{\MT@val}%
4189 \fi
4190 \ifx\@tempa\relax\else
4191 \MT@get@size@
4192 \fi

```

Test whether we finally got a number or dimension so that we can strip the 'pt' (`\@defaultunits` and `\strip@pt` are kernel macros).

```

4193 \MT@ifdimen\MT@val{%
4194 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
4195 \edef\MT@val{\strip@pt\@tempdima}%
4196 }{%
4197 \MT@warning{Could not parse font size ~\MT@val'\MessageBreak
4198 in font set ~\MT@curr@set@name'}%
4199 \let\MT@val\relax
4200 }%

```

4201 }

`\MT@get@size@` The `relsize` solution of parsing `\@setfontsize` does not work with the AMS 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).

```
4202 \def\MT@get@size@{%
4203   \begingroup
4204     \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
4205     \@tempa\@nil
4206 }
```

The `svjour3` class defines the size commands using conditionals; using e-TeX primitives, we close any leftovers here.

```
4207 ^^X\@ifclassloaded{svjour3}{%
4208 ^^X   \def\MT@get@size@{%
4209 ^^X     \@tempcnta=\currentiflevel
4210 ^^X     \MT@get@size@@
4211 ^^X     \MT@loop
4212 ^^X       \ifnum\numexpr\currentiflevel-1>\@tempcnta
4213 ^^X         \csname fi\endcsname
4214 ^^X       \MT@repeat
4215 ^^X     }%
4216 ^^X }%
4217 \let\MT@get@size@\MT@get@size@@
4218 ^^X }
```

`\MT@define@set@key@font`

```
4219 \def\MT@define@set@key@font#1{%
4220   \define@key{MT@#1@set}{font}[]{}%
4221   \MT@get@nc{MT@#1@list@font@\MT@curr@set@name}\@empty
4222   \MT@map@clist@{##1}{%
4223     \def\MT@val{###1}%
4224     \MT@ifstreq\MT@val*{\def\MT@val{*/**/**/*}}\relax
4225     \expandafter\MT@get@font\MT@val///\@nil
4226     \MT@exp@two@n@g@addto@macro
4227       {\csname MT@#1@list@font@\MT@curr@set@name\expandafter\endcsname}%
4228     {\MT@val,}%
4229   }%
4230   \expandafter\g@addto@macro\expandafter\MT@font@sets
4231     \csname MT@#1@list@font@\MT@curr@set@name\endcsname
4232   (debug)\MT@dinfnol{1}{-- font: \nameuse{MT@#1@list@font@\MT@curr@set@name}}%
4233   }%
4234 }
```

`\MT@get@font` Translate any asterisks.

```
4235 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
4236   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
4237   \ifx\MT@val\relax\def\MT@val{0}\fi
4238   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
4239   \let\MT@val\@tempb
4240 }
```

`\MT@get@font@` Helper macro, also used by `\MT@get@font@and@size`.

```
4241 \def\MT@get@font@#1#2#3#4#5#6{%
4242   \let\@tempb\@empty
4243   \def\MT@temp{#1/#2/#3/#4/#5}%
4244   \MT@get@axis{encoding}{#1}%
4245   \MT@get@axis{family}{#2}%
4246   \MT@get@axis{series}{#3}%
4247   \MT@get@axis{shape}{#4}%
4248   \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
4249   \MT@ifempty{#5}{%
```

```

4250 \MT@warn@axis@empty{size}{\string\normalsize}%
4251 \def\MT@val{*}%
4252 }{%
4253 \def\MT@val{#5}%
4254 }%
4255 \MT@get@size
4256 }

```

\MT@get@axis

```

4257 \def\MT@get@axis#1#2{%
4258 \def\MT@val{#2}%
4259 \MT@get@highlevel{#1}%
4260 \MT@ifempty\MT@val{%
4261 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
4262 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
4263 }\relax
4264 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
4265 }

```

\MT@warn@axis@empty

```

4266 \def\MT@warn@axis@empty#1#2{%
4267 \MT@warning{#1 axis is empty in font specification\MessageBreak
4268 \MT@temp'. Using `#2' instead}%
4269 }

```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

```

4270 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
4271 \MT@define@set@key@{encoding}{#1}%
4272 \MT@define@set@key@{family}{#1}%
4273 \MT@define@set@key@{series}{#1}%
4274 \MT@define@set@key@{shape}{#1}%
4275 \MT@define@set@key@size{#1}%
4276 \MT@define@set@key@font{#1}%
4277 }

```

\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.

```

4278 \def\UseMicrotypeSet{%
4279 \MT@begin@catcodes
4280 \MT@UseMicrotypeSet
4281 }

```

\MT@UseMicrotypeSet

```

4282 \newcommand*\MT@UseMicrotypeSet[2][{}]{%
4283 \MT@ifempty{#1}{%
4284 \MT@map@clist@c\MT@features{\begingroup\MT@use@set{##1}{#2}\endgroup}%
4285 }{%
4286 \MT@map@clist@n{#1}{\begingroup
4287 \MT@ifempty{##1}\relax{%
4288 \MT@is@feature{##1}{activation of set `#2'}{%
4289 \MT@exp@one@n\MT@use@set
4290 {\csname MT@rbba@##1\endcsname}{#2}%
4291 }%
4292 }%
4293 \endgroup}%
4294 }%
4295 \MT@end@catcodes
4296 }

```

\MT@pr@setname Only use sets that have been declared.

```

\MT@ex@setname 4297 \def\MT@use@set#1#2{%
\MT@tr@setname 4298 \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
\MT@sp@setname 4299 \MT@xdef@n{MT@#1@setname}{#2}%
\MT@kn@setname
\MT@use@set

```

```

4300 }{%
4301 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
4302 \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
4303 }%
4304 \MT@error{%
4305 The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
4306 Using set ` \@nameuse{MT@#1@setname}' instead}{}%
4307 }%
4308 }

```

`\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.

```

4309 \def\DeclareMicrotypeSetDefault{%
4310 \MT@begin@catcodes
4311 \MT@DeclareMicrotypeSetDefault
4312 }

```

`\MT@DeclareMicrotypeSetDefault`

```

4313 \newcommand*\MT@DeclareMicrotypeSetDefault[2][ ]{%
4314 \MT@ifempty{#1}{%
4315 \MT@map@clist@n{MT@features}{\begingroup\MT@set@default@set{##1}{#2}\endgroup}%
4316 }{%
4317 \MT@map@clist@n{#1}{\begingroup
4318 \MT@ifempty{#1}\relax{%
4319 \MT@is@feature{##1}{declaration of default set `#2'}{%
4320 \MT@exp@one@n\MT@set@default@set
4321 {\csname MT@rba@##1\endcsname}{#2}%
4322 }%
4323 }%
4324 \endgroup}%
4325 }%
4326 \MT@end@catcodes
4327 }

```

`\MT@default@pr@set`

```

\MT@default@ex@set 4328 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 4329 \MT@ifdefined@n@TF{MT@#1@set@#2}{%
\MT@default@sp@set 4330 (debug)\MT@edinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
4331 \MT@xdef@n{MT@default@#1@set}{#2}%
\MT@default@kn@set 4332 }{%
\MT@set@default@set 4333 \MT@error{%
4334 The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
4335 Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
4336 \MT@xdef@n{MT@default@#1@set}{all}%
4337 }%
4338 }

```

1.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version `\MT@variants` appends to the list.

```

4339 \let\MT@variants\@empty
4340 \def\DeclareMicrotypeVariants{%
4341 \MT@begin@catcodes
4342 \@ifstar
4343 \MT@DeclareVariants
4344 {\let\MT@variants\@empty\MT@DeclareVariants}%
4345 }

```

`\MT@DeclareVariants`

```

4346 \def\MT@DeclareVariants#1{%
4347 \MT@map@clist@n{#1}{%
4348 \def\@tempa{##1}%

```

```

4349 \onelevel@sanitize\@tempa
4350 \xdef\MT@variants{\MT@variants{\@tempa}}%
4351 }%
4352 \MT@end@catcodes
4353 }

```

`\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.

```

4354 \def\DeclareMicrotypeAlias{%
4355 \MT@begin@catcodes
4356 \MT@DeclareMicrotypeAlias
4357 }

```

`\MT@DeclareMicrotypeAlias`

```

4358 \newcommand*\MT@DeclareMicrotypeAlias[2]{%
4359 \def\@tempb{#2}%
4360 \onelevel@sanitize\@tempb
4361 \MT@ifdefined@n@T{MT@#1@alias}{%
4362 \MT@warning{Alias font family ` \@tempb' will override
4363 alias ` \@nameuse{MT@#1@alias}'\MessageBreak
4364 for font family `#1'}}%
4365 \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.

```

4366 \MT@ifdefined@c@T\MT@family{%
4367 (debug)\MT@info{1}{Activating alias font ` \@tempb' for ` \MT@family'}%
4368 \MT@glet\MT@familyalias\@tempb
4369 }%
4370 \MT@end@catcodes
4371 }

```

1.3.3 Configuration file management

`\LoadMicrotypeFile` May be used to load a configuration file manually.

```

4372 \def\LoadMicrotypeFile#1{%
4373 \edef\@tempa{\zap@space#1 \empty}%
4374 \onelevel@sanitize\@tempa
4375 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
4376 \ifMT@inlist@
4377 \MT@vinfo{... Configuration file \MT@cfg@prefix-\@tempa.cfg already loaded}%
4378 \else
4379 \MT@xadd\MT@file@list{\@tempa,}%
4380 \MT@begin@catcodes
4381 \InputIfFileExists{\MT@cfg@prefix-\@tempa.cfg}{%
4382 \edef\MT@curr@file{\MT@cfg@prefix-\@tempa.cfg}%
4383 \MT@vinfo{... Loading configuration file \MT@curr@file}%
4384 }{%
4385 \MT@warning{Configuration file \MT@cfg@prefix-\@tempa.cfg\MessageBreak
4386 does not exist}%
4387 }%
4388 \MT@end@catcodes
4389 \fi
4390 }

```

`\MT@cfg@prefix` The configuration files' prefix may be customised.

```

\DeclareMicrotypeFilePrefix 4391 \def\MT@cfg@prefix{mt}
4392 \def\DeclareMicrotypeFilePrefix#1{%
4393 \def\MT@cfg@prefix{#1}%
4394 }
4395 (/package)

```

1.3.4 Disabling ligatures

`\DisableLigatures` 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'.

`\MT@n1@setname` The optional argument may be used to disable selected ligatures only.

```

\MT@n1@ligatures 4396 <pdf-|lua- >
4397 <pdf- >\MT@requires@pdftex5{
4398 \def\DisableLigatures{%
4399 \MT@begin@catcodes
4400 \MT@DisableLigatures
4401 }
4402 \newcommand*\MT@DisableLigatures[2] [] {%
4403 \MT@ifempty{#1}\relax{\gdef\MT@n1@ligatures{#1}}%
4404 \xdef\MT@active@features{\MT@active@features,n1}%
4405 \global\MT@no@ligaturestrue
4406 \MT@declare@sets{n1}{no ligatures}{#2}%
4407 \gdef\MT@n1@setname{no ligatures}%
4408 \MT@end@catcodes
4409 }
4410 <pdf- >{
4411 </pdf-|lua- >

```

If pdfTeX is too old, we throw an error.

```

4412 <pdf-|xe- >
4413 \renewcommand*\DisableLigatures[2] [] {%
4414 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
4415 with pdftex version 1.30 or newer.\MessageBreak
4416 Ignoring \backslashchar DisableLigatures}{%
4417 <pdf- > Upgrade
4418 <xe- > Use
4419 pdftex.}%
4420 }
4421 <pdf- >
4422 </pdf-|xe- >

```

1.3.5 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.

```

4423 <package >
4424 \def\DeclareMicrotypeBabelHook#1#2{%
4425 \MT@map@clist@n{#1}{%
4426 \KV@sp@def\@tempa{##1}%
4427 \MT@gdef@n{\MT@babel@\@tempa}{#2}%
4428 }%
4429 }

```

1.3.6 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).

```

4430 \def\SetProtrusion{%
4431 \MT@begin@catcodes
4432 \MT@SetProtrusion
4433 }

```



```

\MT@SetProtrusion    We want the catcodes to be correct even if this is called in the preamble.
  \MT@pr@c@name 4434 \newcommand*\MT@SetProtrusion[3] [] {%
\MT@extra@context 4435   \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.
                    4436   \MT@set@named@keys{MT@pr@c}{#1}%
                    4437   (debug)\MT@dinfo{1}{creating protrusion list `~\MT@pr@c@name'}%
                    4438   \def\MT@permutelist{pr@c}%
                    4439   \setkeys{MT@cfg}{#2}%

                    We have parsed the second argument, and can now define macros for all permuta-
                    tions of the font attributes to point to \MT@pr@c@{name}, ...
                    4440   \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.
                    4441   \MT@gdef@n{MT@pr@c@~\MT@pr@c@name}{#3}%
                    4442   \MT@end@catcodes
                    4443   }
                    4444   (/package)

\SetExpansion        \SetExpansion only differs in that it allows some extra options (stretch, shrink,
                    step, auto).
                    4445   (*pdf- | lua-)
                    4446   \def\SetExpansion{%
                    4447     \MT@begin@catcodes
                    4448     \MT@SetExpansion
                    4449   }

\MT@SetExpansion
  \MT@ex@c@name 4450 \newcommand*\MT@SetExpansion[3] [] {%
\MT@extra@context 4451   \let\MT@extra@context\@empty
  \MT@permutelist 4452   \MT@set@named@keys{MT@ex@c}{#1}%
                    4453   \MT@ifdefined@n@T{MT@ex@c@~\MT@ex@c@name @factor}{%
                    4454     \ifnum\c@name MT@ex@c@~\MT@ex@c@name @factor\endc@name > \@m
                    4455     \MT@warning@n{Expansion factor \number\@nameuse{MT@ex@c@~\MT@ex@c@name @factor}
                    4456     too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
                    4457     maximum of 1000}%
                    4458     \MT@glet@nc{MT@ex@c@~\MT@ex@c@name @factor}\@m
                    4459     \fi
                    4460   }%
                    4461   (debug)\MT@dinfo{1}{creating expansion list `~\MT@ex@c@name'}%
                    4462   \def\MT@permutelist{ex@c}%
                    4463   \setkeys{MT@cfg}{#2}%
                    4464   \MT@permute
                    4465   \MT@gdef@n{MT@ex@c@~\MT@ex@c@name}{#3}%
                    4466   \MT@end@catcodes
                    4467   }

\SetTracking
                    4468   \def\SetTracking{%
                    4469     \MT@begin@catcodes
                    4470     \MT@SetTracking
                    4471   }

\MT@SetTracking      Third argument may be empty.
                    4472   \newcommand*\MT@SetTracking[3] [] {%
                    4473     \let\MT@extra@context\@empty
                    4474     \MT@set@named@keys{MT@tr@c}{#1}%
                    4475     (debug)\MT@dinfo{1}{creating tracking list `~\MT@tr@c@name'}%
                    4476     \def\MT@permutelist{tr@c}%
                    4477     \setkeys{MT@cfg}{#2}%

```

```

4478 \MT@permute
4479 \KV@sp@def\@tempa{#3}%
4480 \MT@ifempty\@tempa\relax{%
4481   \MT@ifint\@tempa
4482   {\MT@xdef\@n{\MT@tr@cc@\MT@tr@cc@name}{\@tempa}}%
4483   {\MT@warning{Value `@\@tempa' is not a number in\MessageBreak
4484     tracking set `@\MT@curr@set@name'}}}%
4485 \MT@end@catcodes
4486 }
4487 /pdf-|lua-

```

\SetExtraSpacing

```

4488 (*pdf-
4489 \def\SetExtraSpacing{%
4490   \MT@begin@catcodes
4491   \MT@SetExtraSpacing
4492 }

```

\MT@SetExtraSpacing

```

\MT@sp@c@name 4493 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 4494 \let\MT@extra@context\@empty
4495 \MT@set@named@keys{MT@sp@c}{#1}%
\MT@permutelist 4496 (debug)\MT@dinfo{1}{creating spacing list `@\MT@sp@c@name'}%
4497 \def\MT@permutelist{sp@c}%
4498 \setkeys{MT@cfg}{#2}%
4499 \MT@permute
4500 \MT@gdef\@n{\MT@sp@c@\MT@sp@c@name}{#3}%
4501 \MT@end@catcodes
4502 }

```

\SetExtraKerning

```

4503 \def\SetExtraKerning{%
4504   \MT@begin@catcodes
4505   \MT@SetExtraKerning
4506 }

```

\MT@SetExtraKerning

```

\MT@kn@c@name 4507 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 4508 \let\MT@extra@context\@empty
4509 \MT@set@named@keys{MT@kn@c}{#1}%
\MT@permutelist 4510 (debug)\MT@dinfo{1}{creating kerning list `@\MT@kn@c@name'}%
4511 \def\MT@permutelist{kn@c}%
4512 \setkeys{MT@cfg}{#2}%
4513 \MT@permute
4514 \MT@gdef\@n{\MT@kn@c@\MT@kn@c@name}{#3}%
4515 \MT@end@catcodes
4516 }
4517 /pdf-

```

\MT@set@named@keys We first set the name (if specified), then remove it from the list, and set the remaining keys.

\MT@options

```

4518 (*package)
4519 \def\MT@set@named@keys#1#2{%
4520   \def\x##1name=##2,##3\@nil{%
4521     \setkeys{#1}{name=##2}%
4522     \gdef\MT@options{##1##3}%
4523     \MT@rem@from@clist{name=}\MT@options
4524   }%
4525   \x#2,name=,\@nil
4526   \@expandtwoargs\setkeys{#1}\MT@options
4527 }

```

\MT@define@code@key Define the keys for the configuration lists (which are setting the codes, in pdfTeX speak).

```

4528 \def\MT@define@code@key#1#2{%
4529   \define@key{MT@#2}{#1}[]{%
4530     \@tempcnta=\@ne
4531     \MT@map@clist@n{##1}{%
4532       \KV@@sp@def\MT@val{###1}%

```

Here, too, we allow for something like ‘bf*’. It will be expanded immediately.

```

4533     \MT@get@highlevel{#1}%
4534     \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
4535     \advance\@tempcnta \@ne
4536   }%
4537 }%
4538 }

```

`\MT@define@code@key@family` Remove fontspec’s internal feature counter.

```

4539 \def\MT@define@code@key@family#1{%
4540   \define@key{MT@#1}{family}[]{%
4541     \@tempcnta=\@ne
4542     \MT@map@clist@n{##1}{%
4543       \KV@@sp@def\MT@val{###1}%
4544       \MT@get@highlevel{family}%
4545       \ifMT@fontspec
4546         \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
4547       \fi
4548       \MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}%
4549       \advance\@tempcnta \@ne
4550     }%
4551   }%
4552 }

```

`\MT@define@code@key@size` `\MT@tempsize` must be in a `\csname`, so that it is at least `\relax`, not undefined.

```

4553 \def\MT@define@code@key@size#1{%
4554   \define@key{MT@#1}{size}[]{%
4555     \MT@map@clist@n{##1}{%
4556       \KV@@sp@def\MT@val{###1}%
4557       \expandafter\MT@get@range\MT@val--\@nil
4558       \ifx\MT@val\relax \else
4559         \MT@exp@cs\MT@xadd{MT@tempsize}%
4560         {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
4561       \fi
4562     }%
4563   }%
4564 }

```

`\MT@define@code@key@font`

```

4565 \def\MT@define@code@key@font#1{%
4566   \define@key{MT@#1}{font}[]{%
4567     \MT@map@clist@n{##1}{%
4568       \KV@@sp@def\MT@val{###1}%
4569       \MT@ifstreql\MT@val*{\def\MT@val{*/*/*/*/}}\relax
4570       \expandafter\MT@get@font@and@size\MT@val////\@nil
4571       \ifMT@fontspec
4572         \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
4573       \fi
4574       \MT@xdef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
4575       {\csname MT@MT@permutelist @name\endcsname}%
4576 (debug) \MT@dinfo@n1{initialising: use list for font \@tempb=\MT@val}
4577 (debug) \ifx\MT@extra@context\@empty\else\MessageBreak
4578 (debug) (context: \MT@extra@context)\fi}%
4579       \MT@exp@cs\MT@xaddb
4580       {MT@MT@permutelist @\@tempb\MT@extra@context @sizes}%
4581       {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
4582     }%
4583   }%
4584 }

```

`\MT@get@font@and@size` Translate any asterisks and split off the size.

```
4585 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
4586   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
4587 }

4588 \MT@define@code@key{encoding}{cfg}
4589 \MT@define@code@key@family {cfg}
4590 \MT@define@code@key{series} {cfg}
4591 \MT@define@code@key{shape} {cfg}
4592 \MT@define@code@key@size {cfg}
4593 \MT@define@code@key@font {cfg}
```

`\MT@define@opt@key`

```
4594 \def\MT@define@opt@key#1#2{%
4595   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
4596     \MT@xdef@n{MT@#1@c@MT@curr@set@name @#2}{##1}}%
4597 }
```

`\MT@listname@count` The options in the optional first argument.

```
4598 \newcount\MT@listname@count
4599 \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`).

```
4600   \define@key{MT@#1@c}{name}[]{%
4601     \MT@ifempty{##1}%
4602     \MT@ifdefined@nTF{MT@#1@c@MT@curr@file/\the\inputlineno}{%
4603       \global\advance\MT@listname@count\@ne
4604       \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno
4605         (\number\MT@listname@count)}%
4606     }{%
4607       \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
4608     }%
4609   }{%
4610     \MT@edef@n{MT@#1@c@name}{##1}%
4611     \MT@ifdefined@nT{MT@#1@c@csname MT@#1@c@name\endcsname}{%
4612       \MT@warning{Redefining \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}}%
4613     }%
4614   }%
4615   \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
4616 }%
4617 \MT@define@opt@key{#1}{load}%
4618 \MT@define@opt@key{#1}{factor}%
4619 \MT@define@opt@key{#1}{preset}%
4620 \MT@define@opt@key{#1}{inputenc}%
```

Only one context is allowed. This might change in the future.

```
4621   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
4622 }
4623 </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.

```
4624 < *pdf- | lua- >
4625 < pdf- > \MT@requires@pdfTeX7{
4626   \define@key{MT@ex@c}{context}[]{%
4627     \MT@ifempty{##1}\relax{%
4628       \MT@gl@t\MT@copy@font\MT@copy@font@
4629       \def\MT@extra@context{##1}%
4630     }%
4631   }
4632   \MT@addto@setup{%
```

```

4633 \define@key{MT@ex@c}{context}[]{%
4634 \ifx\MT@copy@font\MT@copy@font@
4635 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
4636 \else
4637 \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
4638 Ignoring `context' key\on@line}%
4639 {Either move the settings inside the preamble,\MessageBreak
4640 or load the package with the `copyfonts' option.}%
4641 \fi
4642 }%
4643 }

```

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.

```

4644 \define@key{MT@pr@c}{context}[]{%
4645 \MT@ifempty{#1}\relax{%
4646 \MT@gl@et\MT@copy@font\MT@copy@font@
4647 \def\MT@extra@context{#1}}%
4648 }%
4649 }
4650 \MT@addto@setup{%
4651 \define@key{MT@pr@c}{context}[]{%
4652 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
4653 \ifx\MT@copy@font\MT@copy@font@\else
4654 \MT@warning@n!{If protrusion contexts don't work as expected,
4655 \MessageBreak load the package with the `copyfonts' option}%
4656 \fi
4657 }%
4658 }
4659 </pdf-|lua->
4660 <*pdf->
4661 }{
4662 \define@key{MT@ex@c}{context}[]{%
4663 \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
4664 or later. Ignoring `context' key\on@line}%
4665 {Upgrade pdftex.}%
4666 }
4667 </pdf->
4668 <*pdf-|xe->
4669 \define@key{MT@pr@c}{context}[]{%
4670 \MT@error{Protrusion contexts only work with pdftex
4671 <pdf-> 1.40.4\MessageBreak or later.
4672 <xe-> \MessageBreak or luatex.
4673 Ignoring `context' key\on@line}%
4674 <pdf-> {Upgrade pdftex.}%
4675 <xe-> {Use pdftex or luatex.}%
4676 }
4677 </pdf-|xe->
4678 <pdf->}

```

\MT@warn@nodim

```

4679 <*package>
4680 \def\MT@warn@nodim#1{%
4681 \MT@warning{`\@tempa' is not a dimension.\MessageBreak
4682 Ignoring it and setting values relative to\MessageBreak #1}%
4683 }

```

Protrusion codes may be relative to character width, or to any dimension.

```

4684 \define@key{MT@pr@c}{unit}[character]{%
4685 \MT@gl@et@nc{MT@pr@c@MT@curr@set@name @unit}\@empty

```

```
4686 \def\@tempa{#1}%
4687 \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.

```
4688 \MT@ifdimen\@tempa
4689 {\MT@gletenc{MT@prc@MT@curr@set@name @unit}\@tempa}%
4690 {\MT@warn@nodim{character widths}}%
4691 }%
4692 }
4693 </package>
```

Tracking may only be relative to a dimension.

```
4694 <*pdf-|lua->
4695 \define@key{MT@trc}{unit}[1em]{%
4696 \MT@gletenc{MT@trc@MT@curr@set@name @unit}\@empty
4697 \def\@tempa{#1}%
4698 \MT@ifdimen\@tempa
4699 {\MT@gletenc{MT@trc@MT@curr@set@name @unit}\@tempa}%
4700 {\MT@warn@nodim{1em}%
4701 \MT@gdefn{MT@trc@MT@curr@set@name @unit}{1em}}%
4702 }
4703 </pdf-|lua->
```

Spacing and kerning codes may additionally be relative to space dimensions.

```
4704 <*pdf->
4705 \MT@map@clist@n{sp,kn}{%
4706 \define@key{MT@#1c}{unit}[space]{%
4707 \MT@gletenc{MT@#1c@MT@curr@set@name @unit}\@empty
4708 \def\@tempa{##1}%
4709 \MT@ifstreq\@tempa{character}\relax%
4710 \MT@gletenc{MT@#1c@MT@curr@set@name @unit}\mne
4711 \MT@ifstreq\@tempa{space}\relax%
4712 \MT@ifdimen\@tempa
4713 {\MT@gletenc{MT@#1c@MT@curr@set@name @unit}\@tempa}%
4714 {\MT@warn@nodim{width of space}}%
4715 }%
4716 }%
4717 }%
4718 }
4719 </pdf->
```

The first argument to `\SetExpansion` accepts some more options.

```
4720 <*pdf-|lua->
4721 \MT@map@clist@n{stretch,shrink,step}{%
4722 \define@key{MT@exc}{#1}[]{%
4723 \MT@ifempty{##1}\relax%
4724 \MT@ifint{##1}{%
```

A space terminates the number.

```
4725 \MT@gdefn{MT@exc@MT@curr@set@name @#1}{##1 }%
4726 }%
4727 \MT@warning{%
4728 Value `##1' for option `#1' is not a number.\MessageBreak
4729 Ignoring it}%
4730 }%
4731 }%
4732 }%
4733 }
4734 \define@key{MT@exc}{auto}[true]{%
4735 \def\@tempa{#1}%
4736 \csname if\@tempa\endcsname
```

Don't use `autoexpand` for pdfTeX version older than 1.20.

```
4737 <pdf-> \MT@requires@pdftex4%
```

```

4738 <lua- > \MT@requires@luatex3\relax
4739 { \MT@gdef@n{MT@ex@e@MT@curr@set@name @auto}{autoexpand}}%
4740 <pdf- > { \MT@warning{pdfTeX too old for automatic font expansion}}%
4741 \else
4742 <pdf- > \MT@requires@pdfTeX4%
4743 <*lua- >
4744 \MT@requires@luatex3{%
4745 \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
4746 \luatex}}%
4747 </lua- >
4748 { \MT@glet@nc{MT@ex@e@MT@curr@set@name @auto}\@empty}%
4749 <pdf- > \relax
4750 \fi
4751 }

```

Tracking: Interword spacing and outer kerning. The variant with space just in case `\SetTracking` is called inside an argument (e.g., to `\IfFileExists`).

```

4752 \MT@define@opt@key{tr}{spacing}
4753 \MT@define@opt@key{tr}{outerspacing}
4754 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

4755 \define@key{MT@tr@c}{no ligatures}[]%
4756 { \MT@xdef@n{MT@tr@c@MT@curr@set@name @no ligatures}{#1}}
4757 \define@key{MT@tr@c}{outer spacing}[] { \setkeys{MT@tr@c}{outerspacing=#1}}
4758 \define@key{MT@tr@c}{outer kerning}[] { \setkeys{MT@tr@c}{outerkerning=#1}}
4759 \define@key{MT@tr@c}{no ligatures}[] { \setkeys{MT@tr@c}{no ligatures=#1}}
4760 </pdf-|lua- >

```

1.3.7 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.

```

4761 <*package >
4762 \renewcommand*\DeclareCharacterInheritance[1] [] {%
4763 \let\MT@extra@context\@empty
4764 \let\MT@extra@inputenc\@undefined
4765 \let\MT@inh@feat\@empty
4766 \setkeys{MT@inh@}{#1}%
4767 \MT@begin@catcodes
4768 \MT@set@inh@list
4769 }

```

`\MT@set@inh@list` No need to create an inheritance list for tracking.

```

4770 \def\MT@set@inh@list#1#2{%
4771 \MT@ifempty\MT@inh@feat{%
4772 \MT@map@clist@c\MT@features{\begingroup
4773 \MT@ifstreq{#1}{tr}\relax{\MT@declare@char@inh{#1}{#1}{#2}}%
4774 \endgroup}%
4775 }%
4776 \MT@map@clist@c\MT@inh@feat{\begingroup
4777 \KV@sp@def\@tempa{#1}%
4778 \MT@ifempty\@tempa\relax{%
4779 \edef\@tempa{\csname MT@rbba@\@tempa\endcsname}%
4780 \MT@ifstreq\@tempa{tr}\relax{%

```

```

4781     \MT@exp@one@n\MT@declare@char@inh{\@tempa}{#1}{#2}}%
4782     \endgroup}%
4783   }%
4784   \MT@end@catcodes
4785 }

```

The keys for the optional argument.

```

4786 \MT@map@clist@c\MT@features@long{%
4787   \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}
4788   \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}

```

`\MT@declare@char@inh` The lists cannot be given a name by the user.

```

4789 \def\MT@declare@char@inh#1#2#3{%
4790   \MT@edef@n{MT@#1@inh@name}%
4791   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
4792   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
4793   \MT@if@defined@c@T\MT@extra@inputenc{%
4794     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
4795   <debug>\MT@dinfo{1}{creating inheritance list ` \@nameuse{MT@#1@inh@name}' }%
4796   \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
4797   \def\MT@permutelist{#1@inh}%
4798   \setkeys{MT@inh}{#2}%
4799   \MT@permute
4800 }

```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations. We can reuse the key setup from the configuration lists (`\Set...`).

```

4801 \MT@define@code@key{encoding}{inh}
4802 \MT@define@code@key{family}{inh}
4803 \MT@define@code@key{series}{inh}
4804 \MT@define@code@key{shape}{inh}
4805 \MT@define@code@key{size}{inh}
4806 \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`).

```

4807 \def\MT@inh@do#1,{%
4808   \ifx\relax#1\@empty \else
4809     \MT@inh@split #1==\relax
4810     \expandafter\MT@inh@do
4811   \fi
4812 }

```

`\MT@inh@split` Only gather the inheriting characters here. Their codes will actually be set in `\MT@set@<feature>@codes`.

```

4813 </package>
4814 <*pdf-|lua-|xe->
4815 \def\MT@inh@split#1=#2=#3\relax{%
4816   \def\@tempa{#1}%
4817   \ifx\@tempa\@empty \else
4818     \expandafter\MT@has@inh@prefix\@tempa()\relax\@nil
4819     \MT@get@slot
4820 <pdf-|lua-> \ifnum\MT@char > \m@ne
4821 <xe-> \ifx\MT@char\@empty\else
4822     \let\MT@val\MT@char
4823     \MT@map@clist@n{#2}{%
4824       \def\@tempa{##1}%
4825       \ifx\@tempa\@empty \else
4826         \MT@get@slot

```



```

4827 <pdf-|lua->          \ifnum\MT@char > \m@ne
4828 <xe->                \ifx\MT@char\@empty\else
4829                    \ifx\MT@inh@prefix\@empty
4830                    \MT@exp@cs\MT@xadd\MT@inh@\MT@listname @\MT@val @}{\MT@char}%
4831                    \else
4832                    \MT@exp@cs\MT@xadd\MT@inh@\MT@listname @prefixes}%
4833                    {\MT@val}\MT@char\MT@inh@prefix@}%
4834                    \fi
4835                    \fi
4836                    \fi
4837                    }%
4838 <debug>\MT@info@n1{2}{children of #1 (\MT@val):
4839 <debug> \@nameuse\MT@inh@\MT@listname @\ifx\MT@inh@prefix\@empty\MT@val @\else prefixes\fi}%
4840        \fi
4841        \fi
4842    }
4843 </pdf-|lua-|xe->

```

`\MT@inh@prefix` If the inheriting character is preceded by (`<prefix>`), where `<prefix>` is one of l, r
`\MT@has@inh@prefix` or lr, this has a special meaning for protrusion. For the other features, we ignore these settings.

```

4844 <*package>
4845 \def\MT@has@inh@prefix#1(#2)#3#4\@nil{%
4846     \let\MT@temp\relax
4847     \ifx\relax#3%
4848         \def\@tempa{#1#2}%
4849         \let\MT@inh@prefix\@empty
4850     \else
4851         \MT@ifstreq{\MT@feat}{pr}{%
4852             \MT@ifstreq{#2}{l}{\def\MT@inh@prefix@{\{1000\}{0}\@firstoftwo}{%
4853                 \MT@ifstreq{#2}{r}{\def\MT@inh@prefix@{\{0\}{1000}\@firstoftwo}{%
4854                     \MT@ifstreq{#2}{lr}{\def\MT@inh@prefix@{\{500\}{500}\@firstoftwo}{%
4855                         \MT@warning@n1{`#2' is not a valid prefix in inheritance list%
4856                             \MessageBreak\MT@listname. Ignoring it}%
4857                         \@secondoftwo}}}%
4858                 {\def\@tempa{#3}%
4859                     \def\MT@inh@prefix{#2}%
4860                     \@gobble}%
4861                 {\@firstofone}%
4862                 }{\@firstofone}%
4863                 {\let\MT@char\m@ne
4864                     \let\MT@temp\@gobble
4865                 }%
4866             \fi
4867             \MT@temp
4868         }

```

1.3.8 Permutation

`\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>@/@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:

```
4869 \MT@gdef@n{MT@pr@c@U/euroitc///}{euroitc}
4870 \MT@gdef@n{MT@pr@c@U/euroitcs///}{euroitc}
4871 \MT@gdef@n{MT@pr@c@U/euroitc//it/}{euroitci}
4872 \MT@gdef@n{MT@pr@c@U/euroitcs//it/}{euroitci}
4873 \MT@gdef@n{MT@pr@c@euroitc}{E={100,50}}
4874 \MT@gdef@n{MT@pr@c@euroitci}{E={100,}}
4875 \def\MT@permute{%
4876   \let\MT@cnt@encoding\@ne
4877   \MT@permute@
```

Undefine commands for the next round.

```
4878 \MT@map@tlist@n{{encoding}}{family}{series}{shape}}\MT@permute@reset
4879 \MT@glet\MT@tempsize\@undefined
4880 }
4881 \def\MT@permute@{%
4882   \let\MT@cnt@family\@ne
4883   \MT@permute@@
4884   \MT@increment\MT@cnt@encoding
4885   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
4886   \MT@permute@
4887 }
4888 \def\MT@permute@@{%
4889   \let\MT@cnt@series\@ne
4890   \MT@permute@@@
4891   \MT@increment\MT@cnt@family
4892   \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
4893   \MT@permute@@
4894 }
4895 \def\MT@permute@@@{%
4896   \let\MT@cnt@shape\@ne
4897   \MT@permute@@@@
4898   \MT@increment\MT@cnt@series
4899   \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
4900   \MT@permute@@@@
4901 }
4902 \def\MT@permute@@@@{%
4903   \MT@permute@@@@@
4904   \MT@increment\MT@cnt@shape
4905   \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
4906   \MT@permute@@@@
4907 }
```

\MT@permute@@@@ In order to save some memory, we can ignore unused encodings (inside the document).

```
4908 \def\MT@permute@@@@@{%
4909   \MT@permute@define{encoding}%
4910   \ifMT@document
4911     \ifx\MT@tempencoding\@empty \else
4912       \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
4913       {\expandafter\expandafter\expandafter\@gobble}%
4914     \fi
4915   \fi
4916   \MT@permute@@@@@
4917 }
```

\MT@permute@@@@@

```
4918 \def\MT@permute@@@@@{%
4919   \MT@permute@define{family}%
4920   \MT@permute@define{series}%
4921   \MT@permute@define{shape}%
4922   \edef\@tempa{\MT@tempencoding
4923     /\MT@tempfamily
```

```

4924         /\MT@tempseries
4925         /\MT@tempshape
4926         /\MT@ifdefined@c@T\MT@tempsize *}%

```

Some sanity checks: an encoding must be specified (unless nothing else is).

```

4927 \MT@ifstreq\@tempa{////}\relax{%
4928 \ifx\MT@tempencoding\@empty
4929 \MT@warning{%
4930 You have to specify an encoding for\MessageBreak
4931 \@nameuse{MT@abbr@MT@permutelist} list
4932 ~\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
4933 Ignoring it}%
4934 \else
4935 \MT@ifdefined@c@TF\MT@tempsize{%

```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```

4936 \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
4937 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
4938 }%
4939 \MT@exp@cs\MT@xaddb
4940 {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
4941 \MT@tempsize
4942 <debug>\MT@dinfo@n1{initialising: use list for font \@tempa,\MessageBreak
4943 <debug> sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
4944 <debug> @sizes\endcsname}%
4945 }{%

```

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.

```

4946 \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
4947 \MT@ifstreq{\csname MT@\MT@permutelist @\@tempa\MT@extra@context\endcsname}%
4948 {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname @load\endcsname}%
4949 \relax}%
4950 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
4951 ~\@nameuse{MT@\MT@permutelist @name}' will\MessageBreak override
4952 list ~\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
4953 for \MessageBreak font ~\@tempa'}%
4954 }%
4955 }%
4956 <debug>\MT@dinfo@n1{initialising: use list for font \@tempa
4957 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
4958 <debug> (context: \MT@extra@context)\fi}%
4959 }%
4960 \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
4961 {\csname MT@\MT@permutelist @name\endcsname}%
4962 \fi
4963 }%
4964 }

```

`\MT@permute@define` Define the commands.

```

4965 \def\MT@permute@define#1{%
4966 \@tempcnta=\csname MT@cnt@#1\endcsname\relax
4967 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
4968 {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
4969 {\MT@let@nc{MT@temp#1}\@empty}%
4970 }

```

`\MT@permute@reset` Reset the commands.

```

4971 \def\MT@permute@reset#1{%
4972 \@tempcnta=\@ne
4973 \MT@loop
4974 \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
4975 \advance\@tempcnta\@ne

```

```

4976   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
4977   \iftrue
4978   \iffalse
4979   \MT@repeat
4980 }

\MT@check@rlist   For every new range item in \MT@tempsize, check whether it overlaps with ranges
                  in the existing list.
4981 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

\MT@check@rlist@   Define the current new range and ...
4982 \def\MT@check@rlist@#1#2#3{%
4983   \def\@tempb{#1}%
4984   \def\@tempc{#2}%
4985   \MT@if@false
4986   \MT@exp@cs\MT@map@tlist@c
4987   {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
4988   \MT@check@range
4989 }

\MT@check@range   ... recurse through the list of existing ranges.
4990 \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.
4991 \def\MT@check@range@#1#2#3{%
4992   \MT@ifdim{#2}=\m@ne{%
4993     \MT@ifdim\@tempc=\m@ne{%

      • Both items are simple sizes.

4994       \MT@ifdim\@tempb={#1}\MT@if@true\relax
4995     }{%

      • Item in list is a simple size, new item is a range.

4996       \MT@ifdim\@tempb>{#1}\relax{%
4997         \MT@ifdim\@tempc>{#1}{%
4998           \MT@if@true
4999           \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
5000         }\relax
5001       }%
5002     }%
5003   }{%
5004     \MT@ifdim\@tempc=\m@ne{%

      • Item in list is a range, new item is a simple size.

5005       \MT@ifdim\@tempb<{#2}{%
5006         \MT@ifdim\@tempb<{#1}\relax\MT@if@true
5007       }\relax
5008     }{%

      • Both items are ranges.

5009       \MT@ifdim\@tempb<{#2}{%
5010         \MT@ifdim\@tempc>{#1}{%
5011           \MT@if@true
5012           \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
5013         }\relax
5014       }\relax
5015     }%
5016   }%
5017   \ifMT@if@
5018   \MT@ifstreq{#3}%

```

```

5019     {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
5020     \relax}%
5021     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
5022     ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
5023     list ~#3' for font \@tempa,\MessageBreak size \@tempb}%
5024     }%

```

If we've already found a conflict with this item, we can skip the rest of the list.

```

5025     \expandafter\MT@tlist@break
5026     \fi
5027 }

```

1.4 Package options

1.4.1 Declaring the options

`\ifMT@opt@expansion` Keep track of whether the user explicitly set these options.

```

\ifMT@opt@auto 5028 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 5029 \newif\ifMT@opt@auto
5030 \newif\ifMT@opt@DVI

```

`\MT@optwarn@admissible` Some warnings.

```

5031 \def\MT@optwarn@admissible#1#2{%
5032   \MT@warning@n1{~#1' is not an admissible value for option\MessageBreak
5033     ~#2'. Assuming ~false'}%
5034 }

```

`\MT@optwarn@nan`

```

5035 </package>
5036 <*package|letterspace>
5037 <plain>\MT@requires@l@tex1{
5038   \def\MT@optwarn@nan#1#2{%
5039     \MT@warning@n1{Value ~#1' for option ~#2' is not a\MessageBreak number.
5040       Using default value of \number\@nameuse{MT@#2@default}}%
5041   }
5042   <plain>}\relax
5043   </package|letterspace>
5044   <*package>

```

`\MT@opt@def@set`

```

5045 \def\MT@opt@def@set#1{%
5046   \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}}%
5047   \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
5048   }{%
5049   \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
5050   \MT@warning@n1{The #1 set ~\MT@val' is undeclared.\MessageBreak
5051     Using set ~\@nameuse{MT@\@tempb @setname}' instead}%
5052   }%
5053 }

```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a *set name*).

```

5054 \MT@map@clist@n{protrusion,expansion}{%
5055   \define@key{MT}{#1}[true]}%
5056   \csname MT@opt@#1true\endcsname
5057   \MT@map@clist@n{##1}{%
5058     \KV@sp@def\MT@val{###1}%
5059     \MT@ifempty\MT@val\relax{%
5060       \csname MT@#1true\endcsname
5061       \edef\@tempb{\csname MT@rbb@#1\endcsname}%
5062       \MT@ifstreq\MT@val{true}\relax
5063     }%

```

```

5064 \MT@ifstreq\MT@val{false}{%
5065 \csname MT@#1false\endcsname
5066 }{%
5067 \MT@ifstreq\MT@val{compatibility}{%
5068 \MT@let@nc{MT@\@tempb @level}\@one
5069 }{%
5070 \MT@ifstreq\MT@val{nocompatibility}{%
5071 \MT@let@nc{MT@\@tempb @level}\tw@
5072 }{%

```

If everything failed, it should be a set name.

```

5073 \MT@opt@def@set{#1}%
5074 }%
5075 }%
5076 }%
5077 }%
5078 }%
5079 }%
5080 }%
5081 }

```

activate is a shortcut for protrusion and expansion.

```

5082 \define@key{MT}{activate}[true]{%
5083 \setkeys{MT}{protrusion={#1}}%
5084 \setkeys{MT}{expansion={#1}}%
5085 }

```

spacing, kerning and tracking do not have a compatibility level.

```

5086 \MT@map@clist@n{spacing,kerning,tracking}{%
5087 \define@key{MT}{#1}[true]{%
5088 \MT@map@clist@n{##1}{%
5089 \KV@sp@def\MT@val{###1}%
5090 \MT@ifempty\MT@val\relax{%
5091 \csname MT@#1true\endcsname
5092 \MT@ifstreq\MT@val{true}\relax
5093 }%
5094 \MT@ifstreq\MT@val{false}{%
5095 \csname MT@#1false\endcsname
5096 }{%
5097 \edef\@tempb{\csname MT@rbb@#1\endcsname}%
5098 \MT@opt@def@set{#1}%
5099 }%
5100 }%
5101 }%
5102 }%
5103 }%
5104 }

```

`\MT@def@bool@opt` The true/false options: draft (may be inherited from the class options), auto, selected, babel, DVIoutput, defersetup, copyfonts.

```

5105 \def\MT@def@bool@opt#1#2{%
5106 \define@key{MT}{#1}[true]{%
5107 \def\@tempa{##1}%
5108 \MT@ifstreq\@tempa{true}\relax{%
5109 \MT@ifstreq\@tempa{false}\relax{%
5110 \MT@optwarn@admissible{##1}{#1}%
5111 \def\@tempa{false}%
5112 }%
5113 }%
5114 #2%
5115 }%
5116 }

```

Boolean options that only set the switch.

```

5117 \MT@map@clist@n{draft,selected,babel}{%

```

```

5118 \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
5119 \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.

```

5120 </package>
5121 <pdf-|lua-|xe-
5122 lua->\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
5123 \MT@def@bool@opt{DVIoutput}{%
5124 \csname if\@tempa\endcsname
5125 <pdf-|lua-
5126 \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
5127 \pdfoutput\z@
5128 \else
5129 \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
5130 \pdfoutput\@ne
5131 </pdf-|lua-
5132 xe->\MT@warning@n{Ignoring `DVIoutput' option}%
5133 \fi
5134 }
5135 </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.

```

5136 <*package>
5137 \MT@def@bool@opt{defersetup}{%
5138 \csname if\@tempa\endcsname \else
5139 \AtEndOfPackage{%
5140 \MT@setup@
5141 \let\MT@setup@\empty
5142 \let\MT@addto@setup\@firstofone
5143 }%
5144 \fi
5145 }
5146 </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.

```

5147 <pdf-|lua-
5148 pdf->\MT@requires@pdftex7{
5149 \MT@def@bool@opt{copyfonts}{%
5150 \csname if\@tempa\endcsname
5151 \MT@glet\MT@copy@font\MT@copy@font@
5152 \else
5153 \MT@glet\MT@copy@font\relax
5154 \fi
5155 }
5156 pdf-}{
5157 </pdf-|lua-
5158 <pdf-|xe-
5159 \MT@def@bool@opt{copyfonts}{%
5160 \csname if\@tempa\endcsname
5161 \MT@error
5162 pdf->{The pdftex version you are using is too old\MessageBreak
5163 pdf->to use the `copyfonts' option}{Upgrade pdftex.}%
5164 xe->{The `copyfonts' option does not work with xetex}

```

```

5165 <xe-          {Use pdftex or luatex instead.}%
5166   \fi
5167   }
5168 <pdf-}&
5169 </pdf-|xe-

```

`final` is the opposite to `draft`. It's only kept for backwards compatibility.

```

5170 <*package>
5171 \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.

```

5172 \define@key{MT}{disable}[true]{%
5173   \def\@tempa{#1}%
5174   \MT@ifstreq\@tempa{true}\MT@disabletrue{%
5175     \MT@ifstreq\@tempa{ifdraft}{\ifMT@draft\MT@disabletrue\fi}{%
5176       \MT@ifstreq\@tempa{false}\relax{%
5177         \MT@optwarn@admissible{#1}{disable}%
5178       }%
5179     }%
5180   }%
5181 }

```

For verbose output, we redefine `\MT@vinfo`.

```

5182 \define@key{MT}{verbose}[true]{%
5183   \let\MT@vinfo\MT@info@n1
5184   \def\@tempa{#1}%
5185   \MT@ifstreq\@tempa{true}\relax{%

```

Take problems seriously.

```

5186   \MT@ifstreq\@tempa{errors}{%
5187     \let\MT@warning \MT@warn@err
5188     \let\MT@warning@n1\MT@warn@err
5189   }{%
5190     \let\MT@vinfo\@gobble

```

Cast warnings to the winds.

```

5191   \MT@ifstreq\@tempa{silent}{%
5192     \let\MT@warning \MT@info
5193     \let\MT@warning@n1\MT@info@n1
5194   }{%
5195     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
5196   }%
5197 }%
5198 }%
5199 }
5200 </package>

```

Options with numerical keys: `factor`, `stretch`, `shrink`, `step`, `letterspace`.

```

5201 <*package|letterspace>
5202 <plain>\MT@requires@latex1{
5203 \MT@map@clist@n{%
5204 <package> stretch,shrink,step,%
5205 letterspace}{%
5206 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
5207 \def\@tempa{##1 }%

```

No nonsense in `\MT@factor` et al.? A space terminates the number.

```

5208   \MT@ifint\@tempa
5209   {\MT@edef@n{MT@#1}{\@tempa}}%
5210   {\MT@optwarn@nan{##1}{#1}}%
5211 }%
5212 }
5213 <plain>\relax
5214 </package|letterspace>

```


factor will define the protrusion factor only.

```

5215 (*package)
5216 \define@key{MT}{factor}[\MT@factor@default]{%
5217   \def\@tempa{#1}%
5218   \MT@ifint\@tempa
5219   {\edef\MT@pr@factor{\@tempa}}
5220   {\MT@optwarn@nan{#1}{factor}}%
5221 }

```

Unit for protrusion codes.

```

5222 \define@key{MT}{unit}[character]{%
5223   \def\@tempa{#1}%
5224   \MT@ifstreq\@tempa{character}\relax{%
5225     \MT@ifdimen\@tempa
5226     {\let\MT@pr@unit\@tempa}%
5227     {\MT@warning@n1{\@tempa' is not a dimension.\MessageBreak
5228       Ignoring it and setting values relative to\MessageBreak
5229       character widths}}%
5230   }%
5231 }

```

`\MT@patches@list` The patch and nopatch options. Remember chosen option for later (`\relax` means 'all', `\@empty` means 'none').

```

5232 \let\MT@patches@list\relax
5233 \let\MT@nopatches@list\@empty
5234 \define@key{MT}{patch}[all]{%
5235   \def\@tempa{#1}%
5236   \MT@ifstreq\@tempa{all}
5237   \relax
5238   {\MT@ifstreq\@tempa{none}
5239    {\let\MT@patches@list\@empty}
5240    {\def\MT@patches@list{#1}}}%
5241 }
5242 \define@key{MT}{nopatch}[all]{%
5243   \def\@tempa{#1}%
5244   \MT@ifstreq\@tempa{all}
5245   {\let\MT@nopatches@list\relax}
5246   {\MT@ifstreq\@tempa{none}
5247    \relax
5248    {\def\MT@nopatches@list{#1}}}%
5249 }

```

We can only apply the patches `AtBeginDocument`.

```

5250 \MT@addto@setup{%
5251   \ifx\MT@patches@list\relax
5252     \let\MT@patches@list\MT@patches@def
5253   \fi
5254   \ifx\MT@nopatches@list\@empty\else
5255     \ifx\MT@nopatches@list\relax
5256       \let\MT@nopatches@list\MT@patches@def
5257     \fi
5258     \MT@map@clist@c\MT@nopatches@list{%
5259       \MT@rem@from@clist{#1}\MT@patches@list}%
5260   \fi
5261   \ifx\MT@patches@list\@empty\else
5262     ^X \MT@map@clist@c\MT@patches@list{\MT@apply@patch{#1}}%
5263     ^Q \MT@warning@n1{Patches require the etex extensions. Ignoring them}%
5264   \fi
5265 }

```

1.4.2 Loading the definition file

Load the engine-specific code (as strewn across this file).

```

\MT@get@MT@version    We also check whether versions are the same.
    \MT@version 5266 \def\MT@get@MT@version#1 #2 #3\@nil{#1 #2}
\MT@check@MT@version 5267 \edef\MT@version{\expandafter\expandafter\expandafter\MT@get@MT@version
5268   \csname ver@MT@MT.sty\endcsname\@nil}
5269 \def\MT@check@MT@version#1#2{%
5270   \MT@ifstreq\MT@version{#1}{}%
5271   \MT@warning@nl{Mismatching file versions:\MessageBreak
5272     \MT@MT.sty provides:\MessageBreak` \MT@version',\MessageBreak
5273     whereas #2 provides:\MessageBreak`#1'.\MessageBreak
5274     Please fix your installation}}
5275 \input{\MT@MT-\MT@engine tex.def}
5276 \edef\@tempa{\expandafter\expandafter\expandafter\MT@get@MT@version
5277   \csname ver@MT@MT-\MT@engine tex.def\endcsname\@nil}
5278 \MT@check@MT@version\@tempa{\MT@MT-\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 T_EX systems have switched to the pdfT_EX engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfT_EX.)

```

5279 \MT@protrusiontrue
5280 </package>
5281 < *pdf- |lua->
5282 \ifnum\pdfoutput<\@ne \else

```

Also, we only enable expansion by default if pdfT_EX can expand the fonts automatically.

```

5283 <pdf-> \MT@requires@pdftex4{
5284   \MT@expansiontrue
5285 <pdf-> \MT@autottrue
5286 <pdf-> }\relax
5287 \fi
5288 <lua->\MT@autottrue
5289 </pdf- |lua->

```

The main configuration file will be loaded before processing the package options.

`\MT@config@file` However, the `config` option must of course be evaluated beforehand. We also have to define a no-op for the regular option processing later.

```

5290 < *package>
5291 \define@key{MT}{config}[]{\relax}
5292 \def\MT@temp#1config=#2,#3\@nil{%
5293   \MT@ifempty{#2}%
5294   {\def\MT@config@file{\MT@MT.cfg}}%
5295   {\def\MT@config@file{#2.cfg}}%
5296 }
5297 \expandafter\expandafter\expandafter\MT@temp
5298 \csname opt@\@currname.\@currext\endcsname,config=,\@nil

```

Load the file.

```

5299 \IfFileExists{\MT@config@file}{%
5300   \MT@info@nl{Loading configuration file \MT@config@file}%
5301   \MT@begin@catcodes
5302   \let\MT@begin@catcodes\relax
5303   \let\MT@end@catcodes\relax
5304   \let\MT@curr@file\MT@config@file
5305   \input{\MT@config@file}%
5306   \endgroup
5307 }{\MT@warning@nl{%
5308   Could not find configuration file ` \MT@config@file'!\MessageBreak

```

```

5309 This will almost certainly cause undesired results.\MessageBreak
5310 Please fix your installation}%
5311 }

```

`\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).

```

5312 \def\MTCHECK@ACTIVE@SET#1{%
5313 \MT@ifdefined@n@TF{MT@#1@setname}{%
5314 \MT@info@n1{Using \nameuse{MT@abbr@#1} set `\'@nameuse{MT@#1@setname}'}%
5315 }{%
5316 \MT@ifdefined@n@TF{MT@default@#1@set}{%
5317 \MT@gl@et@nn{MT@#1@setname}{MT@default@#1@set}%
5318 \MT@info@n1{Using default \nameuse{MT@abbr@#1} set `\'@nameuse{MT@#1@setname}'}%
5319 }{%

```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '@', and issue a warning.

```

5320 \MT@gdef@n{MT@#1@setname}{@}%
5321 \MT@warning@n1{No \nameuse{MT@abbr@#1} set chosen, no default set declared.
5322 \MessageBreak Using empty set}%
5323 }%
5324 }%
5325 }

```

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.

```

5326 \MT@ifdefined@c@T\MicroType@Hook{\MT@error{%
5327 Command \@backslashchar MicroType@Hook is deprecated.\MessageBreak
5328 Use \@backslashchar Microtype@Hook instead}
5329 {You might want to inform the font package authors.}\MicroType@Hook}
5330 \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 (unless `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.

```

5331 \def\microtypesetup{\setkeys{MT}}
5332 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
5333 \package
5334 \*pdf-|lua-|xe-
5335 \def\MT@define@optionX#1#2{%
5336   \define@key{MTX}{#1}[true]{%
5337     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
5338     \MT@map@clist@{##1}{%
5339       \KV@sp@def\MT@val{###1}%
5340       \MT@ifempty\MT@val\relax%
5341       \@tempcnta=\m@ne
5342       \MT@ifstreql\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.

```

5343   \MT@checksetup{#1}{%
5344     \@tempcnta=\csname MT@\@tempb @level\endcsname
5345     \MT@vinfo{Enabling #1
5346       (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
5347   }%
5348 }{%
5349   \MT@ifstreql\MT@val{false}{%
5350     \@tempcnta=\z@
5351     \MT@vinfo{Disabling #1\on@line}%
5352   }{%
5353     \MT@ifstreql\MT@val{compatibility}{%
5354       \MT@checksetup{#1}{%
5355         \@tempcnta=\@ne
5356         \MT@let@nc{MT@\@tempb @level}\@ne
5357         \MT@vinfo{Setting #1 to level 1\on@line}%
5358       }%
5359     }{%
5360       \MT@ifstreql\MT@val{nocompatibility}{%
5361         \MT@checksetup{#1}{%
5362           \@tempcnta=\tw@
5363           \MT@let@nc{MT@\@tempb @level}\tw@
5364           \MT@vinfo{Setting #1 to level 2\on@line}%
5365         }%
5366         }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
5367           {Use any of `true', `false', `compatibility' or
5368             `nocompatibility'.}%
5369       }%
5370     }%
5371   }%
5372 }%
5373 \ifnum\@tempcnta>\m@ne
5374   #2\@tempcnta\relax
5375 \fi
5376 }%
5377 }%
5378 }%
5379 }

```

`\MT@checksetup` Test whether the feature wasn't disabled in the package options.

```

5380 \def\MT@checksetup#1{%
5381   \csname ifMT@#1\endcsname

```

```

5382     \expandafter\@firstofone
5383     \else
5384     \MT@error{You cannot enable #1 if it was disabled\MessageBreak
5385             in the package options}{Load microtype with #1 enabled.}%
5386     \expandafter\@gobble
5387     \fi
5388 }

5389 \MT@define@optionX{protrusion}\MT@protrudechars
5390 </pdf-|lua-|xe-
5391 < *pdf-|lua-
5392 \MT@define@optionX{expansion}\MT@adjustspacing

```

\MT@protrudechars

```

\MT@adjustspacing 5393 <*lua-
5394 \MT@requires@luatex4{
5395     \let\pdfprotrudechars\protrudechars
5396     \let\pdfadjustspacing\adjustspacing
5397 } \relax
5398 </lua-
5399 \let\MT@protrudechars\pdfprotrudechars
5400 \let\MT@adjustspacing\pdfadjustspacing
5401 </pdf-|lua-
5402 < *xe-
5403 \let\MT@protrudechars\XeTeXprotrudechars
5404 \define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
5405 </xe-

```

\MT@define@optionX@ The same for tracking, spacing and kerning, which do not have a compatibility level.

```

5406 < *pdf-|lua-
5407 < pdf- \MT@requires@pdftex6{
5408 < lua- \MT@requires@luatex3{
5409     \def\MT@define@optionX@#1#2{%
5410         \define@key{MTX}{#1}[true]{%
5411             \MT@map@clist@n{##1}{%
5412                 \KV@sp@def\MT@val{###1}%
5413                 \MT@ifempty\MT@val\relax{%
5414                     \@tempcnta=\m@ne
5415                     \MT@ifstreq\MT@val{true}{%
5416                         \MT@checksetup{#1}{%
5417                             \@tempcnta=\@ne
5418                             \MT@vinfo{Enabling #1\on@line}%
5419                         }%
5420                     }{%
5421                         \MT@ifstreq\MT@val{false}{%
5422                             \@tempcnta=\z@
5423                             \MT@vinfo{Disabling #1\on@line}%
5424                         }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
5425                             {Use either `true' or `false'}}%
5426                     }%
5427                 }%
5428                 \ifnum\@tempcnta>\m@ne
5429                     #2\relax
5430                 \fi
5431             }%
5432         }%
5433     }%
5434 }

```

We cannot simply let \MT@tracking relax, since this may select the already letter-spaced font instance.

```

5435 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
5436                             \else \let\MT@tracking\MT@tracking@ \fi}
5437 < pdf- \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}

```

```

5438 <pdf- > \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
5439 <pdf- > \pdfappendkern\@tempcnta}
5440 }{
5441 </pdf- |lua- >
5442 <*pdf- |lua- |xe- >

```

Disable for older pdfTeX versions and for XeTeX and LuaTeX.

```

5443 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
5444 <lua- >
5445 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
5446 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
5447 <pdf- >
5448 \define@key{MTX}{activate}[true]{%
5449 \setkeys{MTX}{protrusion=#1}}%
5450 <pdf- |lua- > \setkeys{MTX}{expansion=#1}}%
5451 }
5452 </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.

```

5453 <*package >
5454 \let\MT@saved@setupfont\MT@setupfont
5455 \define@key{MTX}{deactivate}[]{%
5456 \MT@info{Deactivate `MT@MT' package}%
5457 \let\MT@setupfont\relax
5458 }
5459 \define@key{MTX}{reactivate}[]{%
5460 \MT@info{Reactivate `MT@MT' package}%
5461 \let\MT@setupfont\MT@saved@setupfont
5462 }

```

Apply or revert patches.

```

5463 \define@key{MTX}{patch}[all]{%
5464 \def\@tempa{#1}%
5465 \MT@ifstreq\@tempa{all}
5466 {\let\@tempa\MT@patches@def}
5467 {\MT@ifstreq\@tempa{none}
5468 {\let\@tempa\@empty}
5469 \relax}%
5470 \ifx\@tempa\@empty\else
5471 ^^X \MT@map@clist@c\@tempa{\MT@apply@patch{##1}}%
5472 ^^Q \MT@warning@nl{Patches require the etex extensions. Ignoring them}%
5473 \fi
5474 }
5475 \define@key{MTX}{nopatch}[all]{%
5476 \def\@tempa{#1}%
5477 \MT@ifstreq\@tempa{all}
5478 {\let\@tempa\MT@patches@def}
5479 {\MT@ifstreq\@tempa{none}
5480 {\let\@tempa\@empty}
5481 \relax}%
5482 \ifx\@tempa\@empty\else
5483 ^^X \MT@map@clist@c\@tempa{\MT@undo@patch{##1}}%
5484 \fi
5485 }
5486 </package >

```

1.4.6 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

5487 <*package|letterspace>
5488 <plain>\MT@requires@latex1{
5489 \def\MT@ProcessOptionsWithKV#1{%
5490 \let\@tempc\relax
5491 \let\MT@temp\@empty
5492 <plain> \MT@requires@latex2{
5493 \MT@map@clist@c\@classoptionslist{%
5494 \def\CurrentOption{##1}%
5495 \MT@ifdefined@n@T{KV@#1@\expandafter\MT@getkey\CurrentOption=\@nil}{%
5496 \edef\MT@temp{\MT@temp,\CurrentOption,}%
5497 \expandtwoargs\@removeelement\CurrentOption
5498 \@unusedoptionlist\@unusedoptionlist
5499 }%
5500 }%
5501 \edef\MT@temp{\noexpand\setkeys{#1}%
5502 {\MT@temp\@optionlist{\@currname.\@current}}}%

```

plain can handle package options.

```

5503 <*plain>
5504 }{\edef\MT@temp{\noexpand\setkeys{#1}%
5505 {\csname usepkg@options@usepkg@pkg\endcsname}}
5506 </plain>
5507 \MT@temp
5508 \MT@clear@options
5509 }

```

\MT@getkey For key=val in class options.

```

5510 \def\MT@getkey#1=#2\@nil{#1}
5511 \MT@ProcessOptionsWithKV{MT}
5512 <plain>\relax
5513 </package|letterspace>
5514 <*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).

```

5515 \MT@addto@setup{%
5516 \ifMT@disable

```

We disable most of what we've just defined in the 5516 lines above if we are running in disable (aka. draft) mode.

```

5517 \MT@warning@n1{The `disable' option is in effect.\MessageBreak
5518 Disabling all micro-typographic extensions.\MessageBreak
5519 This might lead to different line and page breaks}%
5520 \let\MT@setupfont\relax
5521 \renewcommand*\LoadMicrotypeFile[1]{}%
5522 \renewcommand*\microtypesetup[1]{}%
5523 \renewcommand*\microtypecontext[1]{}%
5524 \renewcommand*\lstyle{}%
5525 \else
5526 \MT@setup@PDF
5527 \MT@setup@copies

```

Fix the font sets.

```

5528 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
5529 \MT@setup@protrusion
5530 \MT@setup@expansion
5531 \MT@setup@tracking
5532 \MT@setup@warntracking
5533 \MT@setup@spacing
5534 \MT@setup@kerning
5535 \MT@setup@noligatures
5536 }
5537 </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!

```
5538 <*/pdf-|lua-|
5539 \def\MT@setup@PDF{%
5540   \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
5541   \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
5542 }
```

`\MT@setup@copies` Working on font copies?

```
5543 \def\MT@setup@copies{%
5544   \ifx\MT@copy@font\relax\else \MT@info@n1{Using font copies for contexts}\fi
5545 }
5546 </pdf-|lua-|
5547 <*/xe-|
5548 \let\MT@setup@PDF\relax
5549 \let\MT@setup@copies\relax
5550 </xe-|
```

`\MT@setup@protrusion` Protrusion.

```
5551 <*/pdf-|lua-|xe-|
5552 \def\MT@setup@protrusion{%
5553   \ifMT@protrusion
5554     \edef\MT@active@features{\MT@active@features,pr}%
5555     \MT@protrudechars\MT@pr@level
5556     \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
5557       \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
5558         factor: \number\MT@pr@factor}\fi
5559     \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit}\fi}%
5560     \MT@check@active@set{pr}%
5561   \else
5562     \let\MT@protrusion\relax
5563     \MT@info@n1{No character protrusion}%
5564   \fi
5565 }
5566 </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.

```
5567 <*/pdf-|lua-|
5568 \def\MT@setup@expansion{%
5569   \ifnum\pdfoutput<\@ne
5570     \ifMT@opt@expansion
5571 <*/lua-|
5572     \ifMT@expansion
5573       \MT@requires@luatex3{%
5574         \MT@warning@n1{Font expansion doesn't work properly with luatex in\MessageBreak
5575           DVI mode: the glyphs won't be actually transformed,\MessageBreak
5576           but will only be shifted. You might want to use\MessageBreak
5577           pdflatex instead. I'll continue anyway ..}%
5578         %\MT@expansionfalse
5579       }\relax
5580     \fi
```



```

5581 {/lua-}
5582   \else
5583     \MT@expansionfalse
5584   \fi
5585 \fi
5586 \ifMT@expansion

```

Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.

```

5587   \ifnum\MT@stretch=\m@ne
5588     \let\MT@stretch\MT@stretch@default
5589   \fi

```

If shrink has not been specified, it will inherit the value from stretch.

```

5590   \ifnum\MT@shrink=\m@ne
5591     \let\MT@shrink\MT@stretch
5592   \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 $\min(\text{stretch}, \text{shrink})/5$, rounded off, minimum value 1.

```

5593   \ifnum\MT@step=\m@ne
5594 {pdf-}   \MT@requires@pdftex6{%
5595     \def\MT@step{1 }%
5596 {*pdf-}
5597   }{%
5598     \ifnum\MT@stretch>\MT@shrink
5599       \ifnum\MT@shrink=\z@
5600         \@tempcnta=\MT@stretch
5601       \else
5602         \@tempcnta=\MT@shrink
5603       \fi
5604     \else
5605       \ifnum\MT@stretch=\z@
5606         \@tempcnta=\MT@shrink
5607       \else
5608         \@tempcnta=\MT@stretch
5609       \fi
5610     \fi
5611     \divide\@tempcnta 5\relax
5612     \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
5613     \edef\MT@step{\number\@tempcnta\space}%
5614   }%
5615 {/pdf-}
5616   \fi
5617   \ifnum\MT@step=\z@
5618     \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
5619       Setting it to one}%
5620     \def\MT@step{1 }%
5621   \fi

```

`\MT@auto` Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the `fix` 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.

```

5622   \let\MT@auto\@empty
5623   \ifMT@auto

```

We turn off automatic expansion if output mode is DVI.

```

5624 <pdf-
5625   \MT@requires@pdftex4{%
5626     \ifnum\pdfoutput<\@ne
5627       \ifMT@opt@auto
5628         \MT@error{%
5629           Automatic font expansion only works for PDF output.\MessageBreak
5630           However, you are creating a DVI file}
5631           {If you have created expanded fonts instances, remove `auto' from%
5632           \MessageBreak the package options. Otherwise, you have to switch
5633           off expansion\MessageBreak completely.}%
5634         \fi
5635         \MT@autofalse
5636       \else
5637         \def\MT@auto{autoexpand}%
5638       \fi

```

Also, if pdfTeX is too old.

```

5639   }{%
5640     \MT@error{%
5641       The pdftex version you are using is too old for\MessageBreak
5642       automatic font expansion}%
5643       {If you have created expanded fonts instances, remove `auto' from\MessageBreak
5644       the package options. Otherwise, you have to switch off expansion\MessageBreak
5645       completely, or upgrade pdftex to version 1.20 or newer.}%
5646     \MT@autofalse
5647     \def\MT@auto{1000 }%
5648   }%
5649 </pdf-
5650 <lua-
5651   \MT@requires@luatex3\relax{\def\MT@auto{autoexpand}}%
5652   \else
5653 </lua-
5654 <pdf-

```

No automatic expansion.

```

5653   \MT@requires@pdftex4\relax{%
5654     \def\MT@auto{1000 }%
5655   }%
5656 </pdf-
5657 <*lua-
5658   \MT@requires@luatex3{%
5659     \ifMT@opt@auto
5660       \MT@error{Non-automatic font expansion does not work with\MessageBreak
5661         luatex}{Remove `auto=false' from the package options, or use pdftex.}%
5662     \MT@autotru
5663     \fi
5664   }\relax
5665 </lua-
5666   \fi

```

Choose the appropriate macro for selected expansion.

```

5667   \ifMT@selected
5668     \let\MT@set@ex@codes\MT@set@ex@codes@s
5669   \else
5670     \let\MT@set@ex@codes\MT@set@ex@codes@n
5671   \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdfTeX error.

```

5672   \ifnum\MT@stretch=\z@
5673     \ifnum\MT@shrink=\z@
5674       \MT@warning@n1{%
5675         Both the stretch and shrink limit are set to zero.\MessageBreak
5676         Disabling font expansion}%
5677       \MT@expansionfalse
5678     \fi
5679   \fi

```

```

5680 \fi
5681 \ifMT@expansion
5682 \edef\MT@active@features{\MT@active@features,ex}%
5683 \MT@adjustspacing\MT@ex@level
5684 \MT@info@n1{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
5685 (level \number\MT@ex@level),\MessageBreak
5686 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
5687 step: \number\MT@step, \ifMT@selected\else non-\fi selected}%

```

`\MT@check@step` Check whether stretch and shrink are multiples of step.

```

5688 \def\MT@check@step##1{%
5689 \@tempcnta=\csname MT@##1\endcsname
5690 \divide\@tempcnta \MT@step
5691 \multiply\@tempcnta \MT@step
5692 \ifnum\@tempcnta=\csname MT@##1\endcsname\else
5693 \MT@warning@n1{The ##1 amount is not a multiple of step.\MessageBreak
5694 The effective maximum ##1 is \the\@tempcnta\space
5695 (step \number\MT@step)}%
5696 \fi
5697 }%
5698 \MT@check@step{stretch}%
5699 \MT@check@step{shrink}%
5700 \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 $X_{\text{Y}}\TeX$, but since expansion doesn't work with $X_{\text{Y}}\TeX$, we don't have to bother.) Since 2019/10/01, the command is robust.

```

5701 \MT@ifdefined@TF{showhyphens }{%
5702 \def\MT@temp##1##2{%
5703 \MT@exp@cs\CheckCommand{showhyphens }[1]{##1}%
5704 \DeclareRobustCommand\showhyphens[1]{##2}}%
5705 }{%
5706 \def\MT@temp##1##2{%
5707 \CheckCommand*\showhyphens[1]{##1}%
5708 \gdef\showhyphens###1{##2}}%
5709 }%
5710 \MT@temp
5711 {\setbox0\vbox{\color@begingroup
5712 \everypar{\parfillskip\z@skip
5713 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
5714 \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}
5715 {\setbox0\vbox{\color@begingroup\pdfadjustspacing\z@
5716 \everypar{\parfillskip\z@skip
5717 \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
5718 \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%
5719 \else
5720 \let\MT@expansion\relax
5721 \MT@info@n1{No font expansion}%
5722 \fi
5723 }
5724 </pdf-|lua-
5725 <*xe-
5726 \def\MT@setup@expansion{%
5727 \ifMT@expansion
5728 \ifMT@opt@expansion
5729 \MT@error{Font expansion does not work with xetex}
5730 {Use pdftex or luatex instead.}%
5731 \fi
5732 \fi
5733 }
5734 </xe-

```

`\MT@setup@tracking` Tracking, spacing and kerning.

```
5735 <pdf-|lua-
5736 <pdf-)>\MT@requires@pdftex6{%
5737 <lua-)>\MT@requires@luatex3{%
5738 \def\MT@setup@tracking{%
5739 \ifMT@tracking
5740 \MT@info@nl{Tracking enabled}%
5741 \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
5742 \ifMT@protrusion\else\MT@protrudechars\@ne\fi
5743 \else
5744 \let\MT@tracking\relax
5745 \MT@info@nl{No adjustment of tracking}%
5746 \fi
5747 }
5748 </pdf-|lua-
```

`\MT@setup@spacing`

```
5749 <pdf-
5750 \def\MT@setup@spacing{%
5751 \ifMT@spacing
5752 \edef\MT@active@features{\MT@active@features,sp}%
5753 \pdfadjustinterwordglue\@ne
5754 \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.

```
5755 \MT@with@package@T{ragged2e}{%
5756 \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
5757 Adjustment of interword spacing may lead to\MessageBreak
5758 undesired results when used with `ragged2e'.\MessageBreak
5759 In this case, disable the `spacing' option}%
5760 }%
5761 \MT@check@active@set{sp}%
5762 \else
5763 \let\MT@spacing\relax
5764 \MT@info@nl{No adjustment of interword spacing}%
5765 \fi
5766 }
```

`\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`.⁹

```
5767 \def\MT@setup@spacing@check{%
5768 \ifMT@spacing
5769 \ifMT@babel \else
5770 \ifnum\sfcode`. > 1500
5771 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
5772 \MT@warning@nl{%
5773 \@backslashchar nonfrenchspacing is active. Adjustment of\MessageBreak
5774 interword spacing will disable it. You might want\MessageBreak
5775 to add \@backslashchar microtypecontext{spacing=nonfrench}'\MessageBreak
5776 to your preamble}%
5777 }%
5778 \fi
5779 \fi
5780 \fi
5781 }
```

`\MT@setup@kerning`

```
5782 \def\MT@setup@kerning{%
```

⁹ Cf. the c.t.t. thread '`\frenchspacing` with AMS packages and babel', started by Philipp Lehman on 16 August 2005, MID: ddtbaj\$rob\$1@online.de

```

5783 \ifMT@kerning
5784 \edef\MT@active@features{\MT@active@features,kn}%
5785 \pdfprependkern\@ne
5786 \pdfappendkern\@ne
5787 \MT@info@n1{Adjustment of character kerning enabled}%
5788 \MT@check@active@set{kn}%
5789 \else
5790 \let\MT@kerning\relax
5791 \MT@info@n1{No adjustment of character kerning}%
5792 \fi
5793 }
5794 </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 XeTeX.

```

5795 <pdf-|lua->{
5796 <*lua->
5797 \def\MT@setup@tracking{%
5798 \ifMT@tracking
5799 \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
5800 or newer. Switching it off}{Upgrade luatex.}%
5801 \MT@trackingfalse
5802 \MT@let@nc{MT@tracking}\relax
5803 \else
5804 \MT@info@n1{No adjustment of tracking (luatex too old)}%
5805 \fi
5806 }
5807 }
5808 </lua->
5809 <*pdf-|lua-|xe->
5810 \def\MT@error@doesnt@work#1{%
5811 \csname ifMT@#1\endcsname
5812 \MT@error{The #1 feature only works with pdftex 1.40\MessageBreak
5813 or newer. Switching it off}
5814 <pdf-> {Upgrade pdftex.}%
5815 <lua-|xe-> {Use pdftex instead.}%
5816 \csname MT@#1false\endcsname
5817 \MT@let@nc{MT@#1}\relax
5818 \else
5819 \MT@info@n1{No adjustment of #1%
5820 <pdf-> \space(pdftex too old)%
5821 }%
5822 \fi
5823 }
5824 <pdf-|xe-> \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
5825 \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
5826 \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
5827 <pdf->}
5828 </pdf-|lua-|xe->

```

`\MT@setup@warntracking`

```

5829 <letterspace>\MT@addto@setup
5830 <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.

```

5831 <*pdf-|lua-|letterspace>
5832 {%
5833 <*pdf-|letterspace>
5834 \ifnum\pdfoutput<\@ne
5835 \def\MT@warn@tracking@DVI{%
5836 <letterspace> \MT@pdf@or@lua{%
5837 \MT@warning@n1{%

```

```

5838         You are using tracking/letterspacing in DVI mode.\MessageBreak
5839         This will probably not work, unless the post-\MessageBreak
5840         processing program (dvips, dvi2pdf(x), ...) is\MessageBreak
5841         able to create the virtual fonts on the fly}%
5842 <letterspace>         }\relax
5843         \MT@gllet\MT@warn@tracking@DVI\relax
5844     }%
5845     \else
5846 </pdf-|letterspace>
5847     \def\MT@warn@tracking@DVI{%
5848         \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
5849         \MT@gllet\MT@warn@tracking@DVI\relax
5850     }%
5851 <pdf-|letterspace> \fi
5852     \ifnum\MT@letterspace=\m@ne
5853         \let\MT@letterspace\MT@letterspace@default
5854     \else
5855         \MT@ls@too@large\MT@letterspace
5856     \fi
5857 }
5858 </pdf-|lua-|letterspace>
5859 <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.

```

5860 <*pdf-|lua->
5861 \def\MT@setup@noligatures{%
5862 <pdf-> \MT@requires@pdftex5{%
5863     \ifMT@noligatures \else
5864         \let\MT@noligatures\relax
5865     \fi
5866 <pdf-> }\relax
5867 }
5868 </pdf-|lua->
5869 <xe->\let\MT@setup@noligatures\relax

```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```

5870 <*package>
5871 \MT@addto@setup{%
5872     \ifx\MT@active@features\@empty \else
5873         \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
5874     \fi
5875     \MT@documenttrue
5876 }

```

`\MT@set@babel@context` Interaction with babel.

```

5877 \def\MT@set@babel@context#1{%
5878     \MT@ifdefined@n@TF{MT@babel@#1}{%
5879         \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
5880         \expandafter\MT@exp@one@n\expandafter\microtypecontext
5881         \csname MT@babel@#1\endcsname
5882     }{%
5883         \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
5884     }%
5885 }

```

`\MT@shorthandoff` Active characters can only be switched off if babel isn't loaded after microtype.

```

5886 \@ifpackageloaded{babel}{
5887     \def\MT@shorthandoff#1#2{%
5888         \MT@info@n1{Switching off #1 babel's active characters (#2)}%
5889         \shorthandoff{#2}}
5890 }{
5891     \def\MT@shorthandoff#1#2{%

```

```

5892 \MT@error{You must load `babel' before `\

```

We patch babel's language switching commands to enable language-dependent setup.

```

5896 \MT@addto@setup{%
5897   \ifMT@babel
5898     \ifpackageloaded{babel}{%
5899       \MT@info@nl{Redefining babel's language switching commands}%
5900       \let\

```

Disable French babel's active characters.

```

5911     \MT@if@false
5912     \MT@with@babel@and@T{french} \MT@if@true
5913     \MT@with@babel@and@T{frenchb} \MT@if@true
5914     \MT@with@babel@and@T{français}\MT@if@true
5915     \MT@with@babel@and@T{canadien}\MT@if@true
5916     \MT@with@babel@and@T{acadian} \MT@if@true
5917     \ifMT@if@\

```

Disable Turkish babel's active characters.

```

5918     \MT@if@false
5919     \MT@with@babel@and@T{turkish} \MT@if@true
5920     \ifMT@if@\

```

In case babel was loaded before microtype:

```

5922     \MT@set@babel@context\language@name

```

The polyglossia package has a useful hook. Unfortunately, compatibility with polyglossia is less useful in itself, as only LuaTeX allows working on font copies, and currently doesn't provide the kerning or spacing feature. But who knows, maybe somebody would want more protrusion in French. . .

```

5923   }{%
5924     \ifpackageloaded{polyglossia}{%
5925       \MT@info@nl{Registering with polyglossia's language switching hook}%
5926       \gappto\polyglossia@language@switched{%
5927         \MT@set@babel@context{\language@name}%
5928       }%
5929       \MT@set@babel@context\language@name
5930     }{%
5931       \MT@warning@nl{%
5932         You did not load the babel or the polyglossia package.\MessageBreak
5933         The `babel' option won't have any effect}%
5934     }%
5935   }%
5936   \fi
5937 }

```

Now we close the \fi from \ifMT@disable.

```

5938 \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.

```
5939 \selectfont}
```

`\MT@curr@file` This is the current file (hopefully with the correct extension).

```
5940 \edef\MT@curr@file{\jobname.tex}
```

```
5941 </package>
```

Finally, execute the setup macro at the end of the preamble, and empty it (the `combine` class calls it repeatedly).

```
5942 <*package|letterspace>
```

```
5943 <plain>\MT@requires@latex1{
```

```
5944 \AtBeginDocument{\MT@setup@ \MT@gl@et\MT@setup@ \@empty}
```

```
5945 <plain>}\relax
```

```
5946 </package|letterspace>
```

Must come at the very, very end.

```
5947 <package>\MT@ifdefined@c@T\MT@setup@spacing@check
```

```
5948 <package> { \AtBeginDocument{\MT@setup@spacing@check} }
```

Restore catcodes.

```
5949 <package|letterspace>\MT@restore@catcodes
```

That was that.

2 Configuration files

Let's now write the font configuration files.

```
5950 (*config)
5951
```

2.1 Font sets

We first declare some sets in the main configuration file.

```
5952 (*m-t)
5953 %%% -----
5954 %%% FONT SETS
5955
5956 \DeclareMicrotypeSet{all}
5957   { }
5958
5959 \DeclareMicrotypeSet{allmath}
5960   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }
5961
5962 \DeclareMicrotypeSet{alltext}
5963   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
5964
5965 \DeclareMicrotypeSet{allmath-nott}
5966   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
5967     family   = {rm*,sf*}
5968   }
5969
5970 \DeclareMicrotypeSet{alltext-nott}
5971   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
5972     family   = {rm*,sf*}
5973   }
5974
5975 \DeclareMicrotypeSet{basicmath}
5976   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
5977     family   = {rm*,sf*},
5978     series   = {md*},
5979     size     = {normalsize,footnotesize,small,large}
5980   }
5981
5982 \DeclareMicrotypeSet{basictext}
5983   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
5984     family   = {rm*,sf*},
5985     series   = {md*},
5986     size     = {normalsize,footnotesize,small,large}
5987   }
5988
5989 \DeclareMicrotypeSet{smallcaps}
5990   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
5991     shape    = {sc*,si,scit}
5992   }
5993
5994 \DeclareMicrotypeSet{footnotesize}
5995   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
5996     size     = {-small}
5997   }
5998
5999 \DeclareMicrotypeSet{scriptsize}
6000   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
```

```

6001     size      = {-footnotesize}
6002   }
6003
6004 \DeclareMicrotypeSet{normal font}
6005   { font = */*/*/*/* }
6006

```

The default sets.

```

6007 %%% -----
6008 %%% DEFAULT SETS
6009
6010 \DeclareMicrotypeSetDefault[protrusion]{alltext}
6011 \DeclareMicrotypeSetDefault[expansion]{alltext-nott}
6012 \DeclareMicrotypeSetDefault[spacing]{alltext-nott}
6013 \DeclareMicrotypeSetDefault[kerning]{alltext}
6014 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
6015

```

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.

```

6016 %%% -----
6017 %%% FONT VARIANTS AND ALIASES
6018
6019 \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_YTeX and LuaTeX in the L^AT_EX format is TU, even if fontspec is not loaded.

```

6020
6021 \MT@if@false
6022 \ifx\UnicodeEncodingName\undefined\else
6023   \MT@if@fstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
6024 \fi
6025 \ifMT@fontspec\MT@if@true\fi
6026 \ifMT@if@
6027 % -- Computer/Latin Modern Roman
6028 \DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
6029   \else
6030 \DeclareMicrotypeAlias{lmr}{cmr}           % lmodern
6031 \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.

```

6032 \DeclareMicrotypeAlias{lmsy}{cmsy}      % ""
6033 \DeclareMicrotypeAlias{lmm}{cmm}        % ""
6034 \DeclareMicrotypeAlias{aer}{cmr}       % ae
6035 \DeclareMicrotypeAlias{zer}{cmr}       % zefonts

```

```

6036 \DeclareMicrotypeAlias{cmor}{cmr} % eco
6037 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty
6038 \DeclareMicrotypeAlias{mlmr}{cmr} % mlmodern
6039 \DeclareMicrotypeAlias{mlmsy}{cmsy} % "
6040 \DeclareMicrotypeAlias{mlmm}{cmm} % "

```

Another, new Computer Modern extension. The `newcomputermodern` package loads it by file name.

```

6041 \DeclareMicrotypeAlias{NewCM10-Book.otf} {New Computer Modern}
6042 \DeclareMicrotypeAlias{NewCM10-Regular.otf}{New Computer Modern}

```

CMU Serif can use the settings from New Computer Modern too.

```

6043 \DeclareMicrotypeAlias{CMU Serif} {New Computer Modern}

```

The packages `pxfonts` and `txfonts` fonts inherit Palatino and Times settings respectively, also the T_EX Gyre fonts Pagella and Termes (formerly: `qfonts`).

```

6044 %% -- Palatino
6045 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
6046 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

```

6047 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
6048 \DeclareMicrotypeAlias{fp9j}{pplj} % "

```

The `newpx` package, a replacement for `pxfonts`.

```

6049 \DeclareMicrotypeAlias{zpllf}{pplx} % newpxtext
6050 \DeclareMicrotypeAlias{zplosf}{pplj} % "
6051 \DeclareMicrotypeAlias{zpltlf}{pplx} % "
6052 \DeclareMicrotypeAlias{zpltosf}{pplj} % "

```

The `domitian` package.

```

6053 \DeclareMicrotypeAlias{Domitian-TLF}{pplx}% domitian
6054 \DeclareMicrotypeAlias{Domitian-T0sF}{pplj}% "

```

The OpenType versions:

```

6055 \DeclareMicrotypeAlias{Palatino Linotype}{Palatino}
6056 \DeclareMicrotypeAlias{Palatino LT Std} {Palatino}
6057 \DeclareMicrotypeAlias{TeX Gyre Pagella} {Palatino}
6058 \DeclareMicrotypeAlias{Domitian} {Palatino}
6059 \DeclareMicrotypeAlias{Asana Math} {Palatino}
6060 %% -- Times New Roman
6061 \DeclareMicrotypeAlias{txr}{ptm} % txfonts

```

The `newtx` package, a replacement for `txfonts`.

```

6062 \DeclareMicrotypeAlias{ntxlf}{ptmx} % newtxtext
6063 \DeclareMicrotypeAlias{ntxtlf}{ptmx} % "
6064 \DeclareMicrotypeAlias{ntxosf}{ptmj} % "
6065 \DeclareMicrotypeAlias{ntxtosf}{ptmj} % "

```

The `tempora` package.

```

6066 \DeclareMicrotypeAlias{Tempora-TLF}{ptmx} % tempora
6067 \DeclareMicrotypeAlias{Tempora-T0sF}{ptmj} % "
6068 \DeclareMicrotypeAlias{qtm}{ptm} % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

The `step` package.

```

6069 \DeclareMicrotypeAlias{STEP-TLF}{ptmx} % step
6070 \DeclareMicrotypeAlias{STEP-T0sF}{ptmj} % "

```

The `stix`, `stix2` and `stickstoo` packages (the latter two have departed a bit from being a Times clone, but still seem close enough).

```

6071 \DeclareMicrotypeAlias{stix}{ptm} % stix
6072 \DeclareMicrotypeAlias{stix2}{ptm} % stix2
6073 \DeclareMicrotypeAlias{SticksTooText-LF}{ptmx}
6074 \DeclareMicrotypeAlias{SticksTooText-TLF}{ptmx}

```

```
6075 \DeclareMicrotypeAlias{SticksTooText-0sF} {ptmj}
6076 \DeclareMicrotypeAlias{SticksTooText-T0sF} {ptmj}
```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (TimesNewRomanMT, TimesNRSevenMT), mtm (TimesSmallTextMT); pte (TimesEuropa); ptt (TimesTen); TimesEighteen; TimesModernEF.

MicroPress's Charter version (chmath).

```
6077 %% -- Charter
6078 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

The XCharter package extends the Charter fonts.

```
6079 \DeclareMicrotypeAlias{XCharter-TLF} {bch} % XCharter
6080 \DeclareMicrotypeAlias{XCharter-T0sF} {bch} % "
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
6081 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
6082 %% -- Garamond
6083 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

The garamondx package, an extension of URW Garamond, providing small caps and oldstyle figures.

```
6084 \DeclareMicrotypeAlias{zgmX}{ugm} % garamondx
6085 \DeclareMicrotypeAlias{zgmj}{ugm} % "
6086 \DeclareMicrotypeAlias{zgmI}{ugm} % "
6087 \DeclareMicrotypeAlias{zgmq}{ugm} % "
```

Because a configuration file for Adobe Garamond wouldn't be permitted for T_EX Live distribution, we use EB Garamond as the base font.

```
6088 \DeclareMicrotypeAlias{pad} {EBGaramond-LF}% Adobe Garamond
6089 \DeclareMicrotypeAlias{padx}{EBGaramond-TLF}% "
6090 \DeclareMicrotypeAlias{padj}{EBGaramond-T0sF}% "
6091 %% --
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
6092 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

The eulervm package virtually extends the Euler fonts.

```
6093 \DeclareMicrotypeAlias{zeur}{eur} % Euler VM
6094 \DeclareMicrotypeAlias{zeus}{eus} % "
```

Euro symbol fonts, to save some files.

```
6095 \DeclareMicrotypeAlias{zpeus} {zpeu} % Adobe Euro sans -> serif
6096 \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.

```
6097 \DeclareMicrotypeAlias{Lato} {TU-basic}
6098 \DeclareMicrotypeAlias{Lato-Regular} {TU-basic}
6099 \DeclareMicrotypeAlias{Fontin} {TU-basic}
6100 \DeclareMicrotypeAlias{Fontin-Regular} {TU-basic}
6101 \DeclareMicrotypeAlias{Bergamo Std} {TU-basic}
```

The fontawesome and fontawesome5 packages are aliased to empty settings (see 3.1.6 and 3.2.6).

```
6102 \DeclareMicrotypeAlias{FontAwesome} {TU-empty} % fontawesome
6103 \DeclareMicrotypeAlias{fontawesomefree} {TU-empty} % fontawesome5
6104 \DeclareMicrotypeAlias{fontawesomepro} {TU-empty}
6105 \DeclareMicrotypeAlias{fontawesomebrands}{TU-empty}
```

6106

2.3 Interaction with babel

Contexts that are to be set when switching to a language.

```

6107 %%% -----
6108 %%% INTERACTION WITH THE `babel' PACKAGE
6109
6110 \DeclareMicrotypeBabelHook
6111   {english,UKenglish,british,USenglish,american}
6112   {kerning=, spacing=nonfrench}
6113
6114 \DeclareMicrotypeBabelHook
6115   {french,français,acadian,canadien}
6116   {kerning=french, spacing=}
6117
6118 \DeclareMicrotypeBabelHook
6119   {turkish}
6120   {kerning=turkish, spacing=}
6121

```

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 $X_{\text{L}}\TeX$ or $\text{Lua}\TeX$, 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 CE for O .

```

6122 </m-t>
6123 <+m-t|ebg|zpeu|mvs>

```

```

6124 %%% -----
6125 %%% CHARACTER INHERITANCE
6126
6127 </m-t|ebg|zpeu|mvs>
6128 <*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.

```

6129 \DeclareCharacterInheritance
6130 { encoding = OT1 }
6131 { f = {011}, % ff
6132   i = {\i},
6133   j = {\j},
6134   O = {\O},
6135   o = {\o}
6136 }
6137

```

2.5.2 T1

Candidates here: 028 ('fi'), 029 ('fl'), 030 ('ffi'), 031 ('ffl'), 156 ('IJ' ligature, since L^AT_EX 2005/12/01 accessible as \IJ), 188 ('ij', \ij), Æ, æ, Œ, œ.

```

6138 \DeclareCharacterInheritance
6139 { encoding = T1 }
6140 { A = {\^A,\A,\^A,\-A,\"A,\r A,\k A,\u A},
6141   a = {\^a,\a,\^a,\-a,\"a,\r a,\k a,\u a},
6142   C = {\^C,\c C,\v C},
6143   c = {\^c,\c c,\v c},
6144   D = {\v D,\DH},
6145   d = {\v d,\dj},
6146   E = {\^E,\E,\^E,\"E,\k E,\v E},
6147   e = {\^e,\e,\^e,\"e,\k e,\v e},
6148   f = {027}, % ff
6149   G = {\u G},
6150   g = {\u g},
6151   I = {\^I,\I,\^I,\"I,\.I},
6152   i = {\^i,\i,\^i,\"i,\i},
6153   j = {\j},
6154   L = {\L,\'L,\v L},
6155   l = {\l,\'l,\v l},
6156   N = {\'N,\-N,\v N},
6157   n = {\'n,\-n,\v n},
6158   O = {\O,\^O,\'O,\^O,\-O,\"O,\H O},
6159   o = {\o,\^o,\'o,\^o,\-o,\"o,\H o},
6160   R = {\'R,\v R},
6161   r = {\'r,\v r},
6162   S = {\'S,\c S,\v S,\SS},
6163   s = {\'s,\c s,\v s},
6164   T = {\c T,\v T},
6165   t = {\c t,\v t},
6166   U = {\^U,\'U,\^U,\"U,\H U,\r U},
6167   u = {\^u,\'u,\^u,\"u,\H u,\r u},
6168   Y = {\'Y,\"Y},
6169   y = {\'y,\"y},
6170   Z = {\'Z,\.Z,\v Z},
6171   z = {\'z,\.z,\v z}

```

The 'soft hyphen' often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```

6172 % - = {127},

```

```
6173 }
6174
```

2.5.3 LY1

More characters: 008 ('fl'), 012 ('fi'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
6175 \DeclareCharacterInheritance
6176 { encoding = LY1 }
6177 { A = {\^A,\'A,\^A,\-A,\"A,\r A},
6178   a = {\^a,\'a,\^a,\-a,\"a,\r a},
6179   C = {\c C},
6180   c = {\c c},
6181   D = {\DH},
6182   E = {\^E,\'E,\^E,\"E},
6183   e = {\^e,\'e,\^e,\"e},
6184   f = {011}, % ff
6185   I = {\^I,\'I,\^I,\"I},
6186   i = {\^i,\'i,\^i,\"i,\i},
6187   L = {\L},
6188   l = {\l},
6189   N = {\-N},
6190   n = {\-n},
6191   O = {\^O,\'O,\^O,\-O,\"O,\O},
6192   o = {\^o,\'o,\^o,\-o,\"o,\o},
6193   S = {\v S},
6194   s = {\v s},
6195   U = {\^U,\'U,\^U,\"U},
6196   u = {\^u,\'u,\^u,\"u},
6197   Y = {\'Y,\"Y},
6198   y = {\'y,\"y},
6199   Z = {\v Z},
6200   z = {\v z}
6201 }
6202
```

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.

```
6203 \DeclareCharacterInheritance
6204 { encoding = OT4 }
6205 { A = {\k A},
6206   a = {\k a},
6207   C = {\'C},
6208   c = {\'c},
6209   E = {\k E},
6210   e = {\k e},
6211   f = {011}, % ff
6212   i = {\i},
6213   j = {\j},
6214   L = {\L},
6215   l = {\l},
6216   N = {\'N},
6217   n = {\'n},
6218   O = {\O,\"O},
6219   o = {\o,\"o},
6220   S = {\'S},
6221   s = {\'s},
6222   Z = {\'Z,\"Z},
6223   z = {\'z,\"z},
6224   \textquotedblleft = "FF
6225 }
6226
```

2.5.5 QX

The Central European QX encoding.¹⁰ Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

6227 \DeclareCharacterInheritance
6228   { encoding = QX }
6229   { A = {\^A,\'A,\^A,\-A,\"A,\k A,\AA},
6230     a = {\`a,\'a,\^a,\-a,\"a,\k a,\aa},
6231     C = {\'C,\c C},
6232     c = {\'c,\c c},
6233     D = {\DH},
6234     E = {\^E,\'E,\^E,\"E,\k E},
6235     e = {\`e,\'e,\^e,\"e,\k e},
6236     f = {011}, % ff
6237     I = {\^I,\'I,\^I,\"I,\k I},
6238     i = {\`i,\'i,\^i,\"i,\k i,\i},
6239     j = {\j},
6240     L = {\L},
6241     l = {\l},
6242     N = {\'N,\-N},
6243     n = {\'n,\-n},
6244     O = {\0,\^0,\'0,\^0,\-0,\"0},
6245     o = {\o,\`o,\'o,\^o,\-o,\"o},

```

The Romanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously¹¹) been included in QX encoding. They are still kept for backwards compatibility.

```

6246   S = {\'S,\c S,\textcommabelow S,\v S},
6247   s = {\'s,\c s,\textcommabelow s,\v s},
6248   T = {\c T,\textcommabelow T},
6249   t = {\c t,\textcommabelow t},
6250   U = {\^U,\'U,\^U,\"U,\k U},
6251   u = {\`u,\'u,\^u,\"u,\k u},
6252   Y = {\'Y,\"Y},
6253   y = {\'y,\"y},
6254   Z = {\'Z,\.Z,\v Z},
6255   z = {\'z,\.z,\v z},
6256   . = \textellipsis
6257 }
6258

```

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.

```

6259 \DeclareCharacterInheritance
6260   { encoding = T5 }
6261   { A = {\^A,\'A,\-A,\h A,\d A,\^A,\u A,
6262         \^{\Acircumflex,\'\Acircumflex,\-\Acircumflex,\h\Acircumflex,\d\Acircumflex},
6263         \^{\Abreve,\'\Abreve,\-\Abreve,\h\Abreve,\d\Abreve},
6264     a = {\`a,\'a,\-a,\h a,\d a,\^a,\u a,
6265         \^{\acircumflex,\'\acircumflex,\-\acircumflex,\h\acircumflex,\d\acircumflex},
6266         \^{\abreve,\'\abreve,\-\abreve,\h\abreve,\d\abreve},
6267     D = {\DJ},
6268     d = {\dj},
6269     E = {\^E,\'E,\-E,\h E,\d E,\^E,
6270         \^{\Ecircumflex,\'\Ecircumflex,\-\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
6271     e = {\`e,\'e,\-e,\h e,\d e,\^e,
6272         \^{\ecircumflex,\'\ecircumflex,\-\ecircumflex,\h\ecircumflex,\d\ecircumflex},

```

¹⁰ Contributed by *Maciej Eder*.

¹¹ Cf. <https://tug.org/pipermail/tex-live/2008-August/017204.html>


```

6273 I = {\^I,\'I,\-I,\h I,\d I},
6274 i = {\^i,\'i,\-i,\h i,\d i,\i},
6275 O = {\^O,\'O,\-O,\h O,\d O,\^O,\horn O,
6276   \^Ocircumflex,\'Ocircumflex,\-Ocircumflex,\hOcircumflex,\dOcircumflex,
6277   \^Ohorn,\'Ohorn,\-Ohorn,\hOhorn,\dOhorn},
6278 o = {\^o,\'o,\-o,\h o,\d o,\^o,\horn o,
6279   \^ocircumflex,\'ocircumflex,\-ocircumflex,\hocircumflex,\docircumflex,
6280   \^ohorn,\'ohorn,\-ohorn,\hohorn,\dohorn},
6281 U = {\^U,\'U,\-U,\h U,\d U,\horn U,
6282   \^Uhorn,\'Uhorn,\-Uhorn,\hUhorn,\dUhorn},
6283 u = {\^u,\'u,\-u,\h u,\d u,\horn u,
6284   \^uhorn,\'uhorn,\-uhorn,\huhorn,\duhorn},
6285 Y = {\^Y,\'Y,\-Y,\h Y,\d Y},
6286 y = {\^y,\'y,\-y,\h y,\d y}
6287 }
6288

```

2.5.7 EU1, EU2, TU

The EU1 (X_YTeX), 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.

```

6289 \DeclareCharacterInheritance
6290 { encoding = {TU,EU1,EU2} }
6291 { A = {\^A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
6292   a = {\^a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
6293   C = {\'C,\c C,\v C},
6294   c = {\'c,\c c,\v c},
6295   D = {\v D,\DH},
6296   d = {\v d,\dj},
6297   E = {\^E,\'E,\^E,\^E,\k E,\v E},
6298   e = {\^e,\'e,\^e,\^e,\k e,\v e},
6299 %   f = {/f_f}, % sometimes /f_f, sometimes /ff
6300   G = {\u G},
6301   g = {\u g},
6302   I = {\^I,\'I,\^I,\^I,\"I,\.I},
6303   i = {\^i,\'i,\^i,\^i,\"i,\.i},
6304 %   j = {\j},
6305   L = {\L,\'L,\v L},
6306   l = {\l,\'l,\v l},
6307   N = {\'N,\-N,\v N},
6308   n = {\'n,\-n,\v n},
6309   O = {\^O,\^O,\'O,\^O,\-O,\"O,\H O},
6310   o = {\^o,\^o,\'o,\^o,\-o,\"o,\H o},
6311   R = {\'R,\v R},
6312   r = {\'r,\v r},
6313   S = {\'S,\c S,\v S}, % \SS
6314   s = {\'s,\c s,\v s},
6315   T = {\c T,\v T},
6316   t = {\c t,\v t},
6317   U = {\^U,\'U,\^U,\^U,\H U,\r U},
6318   u = {\^u,\'u,\^u,\^u,\H u,\r u},
6319   Y = {\'Y,\"Y},
6320   y = {\'y,\"y},
6321   Z = {\'Z,\.Z,\v Z},
6322   z = {\'z,\.z,\v z}
6323 }
6324
6325 </m-t>

```

2.5.8 LGR

The Greek LGR encoding. EB Garamond contains some more glyphs.

```

6326 <*-t|ebg>
6327 \DeclareCharacterInheritance
6328   { encoding = LGR,
6329   <ebg>   family = {EBGaramond-OsF,EBGaramond-TosF,EBGaramond-LF,EBGaramond-TLF}
6330   }
6331   {
6332   <m-t>   A = {012},
6333   <ebg>   A = {009,012,253},
6334   <ebg> (1)E = {199},
6335   <ebg>   H = {010},
6336   <ebg> (1)H = {159},
6337   I = {219},
6338   <ebg> (1)I = {155},
6339   O = J,
6340   <ebg> (1)O = {151},
6341   U = {013,223},
6342   W = {011},
6343   a = {014,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,
6344       144,145,146,148,149,150,248},
6345   e = {224,225,226,227,232,233,234,235},
6346   h = {152,153,154,156,157,158,160,161,162,163,164,165,166,167,168,169,170,
6347       171,172,173,174,175,249},
6348   <m-t>   i = {200,201,202,203,208,209,210,211,216,217,218,240,241,242,243},
6349   <ebg>   i = {008,200,201,202,203,208,209,210,211,216,217,218,240,241,242,243},
6350   o = {228,229,230,231,236,237,238,239},
6351   r = {251,252},
6352   u = {015,204,205,206,207,212,213,214,215,220,221,222,244,245,246,247},
6353   w = {176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,
6354       193,194,196,197,198,250},
6355   <ebg>   \textstigma = \textvarstigma,
6356   . = {059} % ano teleia
6357   }
6358
6359 </m-t|ebg>

```

2.5.9 Euro symbols

Make Euro symbols settings simpler.

```

6360 <*-zpeu>
6361 \DeclareCharacterInheritance
6362   { encoding = U,
6363   family = {zpeu,zpeus,eurosans} }
6364   { E = 128 }
6365
6366 </zpeu>
6367 <*-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.

```

6368 \DeclareCharacterInheritance
6369   { encoding = {OT1,U},
6370   family = mvs }
6371   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
6372
6373 </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.

```
6374 (*m-t)
6375 %%% -----
6376 %%% TRACKING/LETTERSPACING
6377
6378 \SetTracking
6379 [ name = default,
6380   no ligatures = {f} ]
6381 { encoding = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
6382 { }
6383
```

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).

```
6384 %%% -----
6385 %%% EXPANSION
6386
6387 \SetExpansion
6388 [ name = default ]
6389 { encoding = {OT1,OT4,QX,T1,LY1} }
6390 {
6391   A = 500,    a = 700,
6392   \AE = 500,  \ae = 700,
6393   B = 700,    b = 700,
6394   C = 700,    c = 700,
6395   D = 500,    d = 700,
6396   E = 700,    e = 700,
6397   F = 700,
6398   G = 500,    g = 700,
6399   H = 700,    h = 700,
6400   K = 700,    k = 700,
6401   M = 700,    m = 700,
6402   N = 700,    n = 700,
6403   O = 500,    o = 700,
6404   \OE = 500,  \oe = 700,
6405   P = 700,    p = 700,
6406   Q = 500,    q = 700,
6407   R = 700,
6408   S = 700,    s = 700,
6409   U = 700,    u = 700,
6410   W = 700,    w = 700,
6411   Z = 700,    z = 700,
6412   2 = 700,
6413   3 = 700,
6414   6 = 700,
6415   8 = 700,
6416   9 = 700
6417 }
6418
```

Settings for Cyrillic T2A encoding.¹²

```
6419 \SetExpansion
6420 [ name = T2A ]
6421 { encoding = T2A }
6422 {
6423   A = 500,    a = 700,
```

¹² Contributed by *Karl Karlsson*.

```

6424     B = 700,      b = 700,
6425     C = 700,      c = 700,
6426     D = 500,      d = 700,
6427     E = 700,      e = 700,
6428     F = 700,
6429     G = 500,      g = 700,
6430     H = 700,      h = 700,
6431     K = 700,      k = 700,
6432     M = 700,      m = 700,
6433     N = 700,      n = 700,
6434     O = 500,      o = 700,
6435     P = 700,      p = 700,
6436     Q = 500,      q = 700,
6437     R = 700,
6438     S = 700,      s = 700,
6439     U = 700,      u = 700,
6440     W = 700,      w = 700,
6441     Z = 700,      z = 700,
6442     2 = 700,
6443     3 = 700,
6444     6 = 700,
6445     8 = 700,
6446     9 = 700,
6447     \CYRA = 500,    \cyra = 700,
6448     \CYRB = 700,    \cyrb = 700,
6449     \CYRV = 700,    \cyrv = 700,
6450     \CYRG = 700,    \cyrg = 700,
6451     \CYRD = 700,    \cyrd = 700,
6452     \CYRE = 700,    \cyre = 700,
6453     \CYRZH = 700,   \cyrzh = 700,
6454     \CYRZ = 700,    \cyrz = 700,
6455     \CYRI = 700,    \cyri = 700,
6456     \CYRISHRT = 700, \cyrishrt = 700,
6457     \CYRK = 700,    \cyrk = 700,
6458     \CYRL = 700,    \cyrl = 700,
6459     \CYRM = 700,    \cyrm = 700,
6460     \CYRN = 700,    \cyrn = 700,
6461     \CYRO = 500,    \cyro = 700,
6462     \CYRP = 700,    \cyrp = 700,
6463     \CYRR = 700,    \cyrr = 700,
6464     \CYRS = 700,    \cyrs = 700,
6465     \CYRT = 700,    \cyrt = 700,
6466     \CYRU = 700,    \cyru = 700,
6467     \CYRF = 700,    \cyrf = 700,
6468     \CYRH = 700,    \cyrh = 700,
6469     \CYRC = 700,    \cyrc = 700,
6470     \CYRCH = 700,   \cyrch = 700,
6471     \CYRSH = 700,   \cyrsh = 700,
6472     \CYRSHCH = 700,  \cyrshch = 700,
6473     \CYRHRSN = 700,  \cyrhdsn = 700,
6474     \CYRERY = 700,   \cyrery = 700,
6475     \CYRSFTSN = 700, \cyrsftsn = 700,
6476     \CYREREV = 700,  \cyrerev = 700,
6477     \CYRYU = 700,    \cyryu = 700,
6478     \CYRYA = 700,    \cyrya = 700
6479   }
6480

```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```

6481 \SetExpansion
6482   [ name = T5 ]
6483   { encoding = T5 }
6484   {
6485     A = 500,      a = 700,
6486     B = 700,      b = 700,

```

```

6487 C = 700, c = 700,
6488 D = 500, d = 700,
6489 E = 700, e = 700,
6490 F = 700,
6491 G = 500, g = 700,
6492 H = 700, h = 700,
6493 K = 700, k = 700,
6494 M = 700, m = 700,
6495 N = 700, n = 700,
6496 O = 500, o = 700,
6497 P = 700, p = 700,
6498 Q = 500, q = 700,
6499 R = 700,
6500 S = 700, s = 700,
6501 U = 700, u = 700,
6502 W = 700, w = 700,
6503 Z = 700, z = 700,
6504 2 = 700,
6505 3 = 700,
6506 6 = 700,
6507 8 = 700,
6508 9 = 700
6509 }
6510
6511 </m-t>

```

2.8 Character protrusion

```

6512 %%% -----
6513 %%% PROTRUSION
6514

```

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.

```
6515 <*cfg-t>
6516 \SetProtrusion
6517 <m-t> [ name = default ]
```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```
6518 <bch> [ name = bch-default ]
```

- Bitstream Letter Gothic (blg)

```
6519 <blg> [ name = blg-default ]
```

- Computer Modern Roman (cmr)

```
6520 <cmr> [ name = cmr-default ]
```

- EB Garamond

```
6521 <ebg> [ name = EBGaramond-default ]
```

- Minion¹³ (pmnx, pmnj)

```
6522 <pmn> [ name = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
6523 <ppl> [ name = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
6524 <ptm> [ name = ptm-default ]
```

- URW Garamond (ugm)

```
6525 <ugm> [ name = ugm-default ]
6526 <m-t|cmr|pmn|ebg> { }
6527 <bch|blg|ugm> { encoding = OT1,
6528 <ppl|ptm> { encoding = {OT1,OT4},
6529 <bch> family = bch }
6530 <blg> family = blg }
6531 <ppl> family = {ppl,pplx,pplj} }
6532 <ptm> family = {ptm,ptmx,ptmj} }
6533 <ugm> family = ugm }
6534 {
6535 <m-t|bch|blg|cmr|ebg|pmn|ppl|ptm> A = {50,50},
6536 <ugm> A = {50,100},
6537 <ebg|ptm> \AE = {50, },
6538 <ugm> \AE = {150,50},
6539 <ugm> B = { ,50},
6540 <bch|ebg|pmn|ugm> C = {50, },
6541 <bch|ebg|pmn> D = { ,50},
6542 <ugm> D = { ,70},
6543 <ugm> E = { ,50},
6544 <m-t|bch|cmr|ebg|pmn|ptm> F = { ,50},
6545 <ugm> F = { ,70},
6546 <bch|ebg|pmn> G = {50, },
6547 <ugm> G = {50,50},
6548 <blg> I = {150,150},
6549 <m-t|cmr|ebg|pmn|ppl|ptm|ugm> J = {50, },
6550 <bch|blg> J = {100, },
```

6551 $\langle !blg \rangle$ K = { ,50},
6552 $\langle blg \rangle$ K = {50, },
6553 $\langle m-t|bch|cmr|ebg|pmn|ppl \rangle$ L = { ,50},
6554 $\langle blg \rangle$ L = { ,150},
6555 $\langle ptm \rangle$ L = { ,80},
6556 $\langle ugm \rangle$ L = { ,120},
6557 $\langle bch|ebg|pmn|ugm \rangle$ O = {50,50},
6558 $\langle ebg \rangle$ \OE = {50, },
6559 $\langle ugm \rangle$ \OE = {50,50},
6560 $\langle blg \rangle$ P = { ,100},
6561 $\langle ugm \rangle$ P = { ,50},
6562 $\langle bch|ebg|pmn \rangle$ Q = {50,70},
6563 $\langle ugm \rangle$ Q = {50,50},
6564 $\langle bch \rangle$ R = { ,50},
6565 $\langle ugm|ebg \rangle$ R = { ,70},
6566 $\langle m-t|bch|cmr|pmn|ppl|ptm \rangle$ T = {50,50},
6567 $\langle blg \rangle$ T = {100,100},
6568 $\langle ebg|ugm \rangle$ T = {70,70},
6569 $\langle m-t|bch|cmr|ebg|pmn|ppl|ptm \rangle$ V = {50,50},
6570 $\langle blg|ugm \rangle$ V = {70,70},
6571 $\langle m-t|bch|cmr|ebg|pmn|ppl|ptm \rangle$ W = {50,50},
6572 $\langle ugm \rangle$ W = {70,70},
6573 $\langle m-t|bch|cmr|ebg|pmn|ppl|ptm \rangle$ X = {50,50},
6574 $\langle ugm \rangle$ X = {50,70},
6575 $\langle m-t|bch|cmr|ebg|pmn|ppl \rangle$ Y = {50,50},
6576 $\langle blg|ptm|ugm \rangle$ Y = {80,80},
6577 $\langle ugm \rangle$ Z = {50,50},
6578 $\langle blg \rangle$ f = {150,100},
6579 $\langle blg \rangle$ i = {150,150},
6580 $\langle blg \rangle$ j = {100,100},
6581 $\langle m-t|bch|cmr|ebg|pmn|ppl|ptm \rangle$ k = { ,50},
6582 $\langle ugm \rangle$ k = { ,70},
6583 $\langle blg \rangle$ l = {150,150},
6584 $\langle pmn \rangle$ l = { , -50},
6585 $\langle ppl \rangle$ p = {50,50},
6586 $\langle ebg|ugm \rangle$ p = { ,50},
6587 $\langle ebg|ppl \rangle$ q = {50, },
6588 $\langle !blg \rangle$ r = { ,50},
6589 $\langle blg \rangle$ r = {100, 80},
6590 $\langle cmr|ebg|pmn \rangle$ t = { ,70},
6591 $\langle bch \rangle$ t = { ,50},
6592 $\langle blg \rangle$ t = {150, 80},
6593 $\langle ugm \rangle$ t = { ,100},
6594 $\langle m-t|bch|cmr|ebg|pmn|ppl|ptm \rangle$ v = {50,50},
6595 $\langle blg \rangle$ v = {100,100},
6596 $\langle ugm \rangle$ v = {50,70},
6597 $\langle m-t|bch|cmr|ebg|pmn|ppl|ptm \rangle$ w = {50,50},
6598 $\langle ugm \rangle$ w = {50,70},
6599 $\langle !blg \rangle$ x = {50,50},
6600 $\langle blg \rangle$ x = {100,100},
6601 $\langle m-t|bch|ebg|pmn \rangle$ y = { ,50},
6602 $\langle blg \rangle$ y = { 50,100},
6603 $\langle cmr|ppl|ptm \rangle$ y = {50,70},
6604 $\langle ugm \rangle$ y = { ,70},

6605 $\langle cmr \rangle$ 0 = { ,50},
6606 $\langle m-t \rangle$ 1 = {50,50},
6607 $\langle bch|blg|ptm|ugm \rangle$ 1 = {150,150},
6608 $\langle cmr \rangle$ 1 = {100,200},
6609 $\langle pmn \rangle$ 1 = { ,50},
6610 $\langle ppl \rangle$ 1 = {100,100},
6611 $\langle bch|cmr|ugm \rangle$ 2 = {50,50},
6612 $\langle blg \rangle$ 2 = { ,100},
6613 $\langle bch|pmn \rangle$ 3 = {50, },
6614 $\langle cmr|ugm \rangle$ 3 = {50,50},
6615 $\langle blg \rangle$ 3 = {100, },

```

6616 <m-t>      4 = {50,50},
6617 <bch>      4 = {100,50},
6618 <blg>      4 = {100, },
6619 <cmr|ugm>   4 = {70,70},
6620 <pmn>      4 = {50, },
6621 <ptm>      4 = {70, },
6622 <cmr>      5 = { ,50},
6623 <bch>      6 = {50, },
6624 <cmr>      6 = { ,50},
6625 <m-t>      7 = {50,50},
6626 <bch|pmn|ugm> 7 = {50,80},
6627 <blg>      7 = {100,100},
6628 <cmr|ptm>   7 = {50,100},
6629 <ppl>      7 = { ,50},
6630 <cmr>      8 = { ,50},
6631 <bch>      9 = {50,50},
6632 <cmr>      9 = { ,50},
6633 <m-t|cmr|pmn|ppl|ptm|ugm> . = { ,700},
6634 <bch|ebg>   . = { ,600},
6635 <blg>      . = {400,500},
6636 <!blg>     {,}= { ,500},
6637 <blg>     {,}= {300,400},
6638 <m-t|cmr|pmn|ppl|ptm|ugm> : = { ,500},
6639 <bch|ebg>   : = { ,400},
6640 <blg>      : = {300,400},
6641 <m-t|bch|ebg|pmn|ptm> ; = { ,300},
6642 <blg>     ; = {200,300},
6643 <cmr|ppl>  ; = { ,500},
6644 <ugm>     ; = { ,400},
6645 <!blg>    ! = { ,100},
6646 <blg>    ! = {200,200},
6647 <m-t|ebg|pmn|ptm> ? = { ,100},
6648 <bch|cmr|ppl|ugm> ? = { ,200},
6649 <blg>     ? = {150,150},
6650 <pmn>    " = {300,300},
6651 <m-t|bch|cmr|ebg|pmn|ppl> @ = {50,50},
6652 <ptm>    @ = {100,100},
6653 <m-t|bch|blg|cmr|ebg|pmn|ppl|ptm> ~ = {200,250},
6654 <ugm>    ~ = {300,350},
6655 <ebg|ppl|ptm> & = {50,100},
6656 <ugm>    & = { ,100},
6657 <m-t|cmr|ebg|pmn> \% = {50,50},
6658 <bch>    \% = { ,50},
6659 <ppl|ptm> \% = {100,100},
6660 <ugm>    \% = {50,100},
6661 <blg>    \# = {100,100},
6662 <m-t|ppl|ptm|ugm> * = {200,200},
6663 <bch|pmn> * = {200,300},
6664 <blg>    * = {150,200},
6665 <cmr|ebg> * = {300,300},
6666 <m-t|cmr|ebg|ppl|ptm> + = {250,250},
6667 <bch>    + = {150,250},
6668 <blg|pmn> + = {150,200},
6669 <ugm>    + = {250,300},
6670 <blg|ugm> {=} = {200,200},
6671 <m-t|ebg|pmn|ptm> ( = {100, }, ) = { ,200},
6672 <bch|ugm>  ( = {200, }, ) = { ,200},
6673 <cmr|blg> ( = {300, }, ) = { ,300},
6674 <ppl>     ( = {100, }, ) = { ,300},
6675 <bch|pmn> [ = {100, }, ] = { ,100},
6676 <blg>    [ = {300,100}, ] = { ,300},

6677 <m-t|ebg|pmn|ptm> / = {100,200},
6678 <bch>    / = { ,200},
6679 <blg>    / = {300,300},
6680 <cmr|ppl> / = {200,300},

```



```

6681 <ugm> / = {100,300},
6682 <m-t|ptm> - = {500,500},
6683 <bch|cmr|ppl> - = {400,500},
6684 <blg> - = {300,400},
6685 <ebg> - = {300,500},
6686 <pmn> - = {200,400},
6687 <ugm> - = {500,600},
6688 <blg> < = {200,100}, > = {100,200},
6689 <blg> _ = {150,250},
6690 <blg> | = {250,250},
6691 <m-t|pmn> \textendash = {200,200}, \textemdash = {150,150},
6692 <bch> \textendash = {200,300}, \textemdash = {150,250},
6693 <cmr> \textendash = {400,300}, \textemdash = {300,200},
6694 <ebg|ppl|ptm> \textendash = {300,300}, \textemdash = {200,200},
6695 <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).

```

6696 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
6697 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
6698 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
6699 <ebg> \textquoteleft = {300,500}, \textquoteright = {400,400},
6700 <ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
6701 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
6702 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
6703 <m-t|ebg|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
6704 <blg> \textquotedblright = {300,400}
6705 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
6706 <ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
6707 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
6708 }
6709

```

Greek uppercase letters are in OT1 encoding only.

```

6710 <*m-t|cmr|ebg|pmn>
6711 \SetProtrusion
6712 <m-t> [ name = OT1-default,
6713 <cmr> [ name = cmr-OT1,
6714 <ebg> [ name = EBGaramond-OT1,
6715 <pmn> [ name = pmnj-OT1,
6716 <m-t> load = default ]
6717 <cmr> load = cmr-default ]
6718 <ebg> load = EBGaramond-default ]
6719 <pmn> load = pmnj-default ]
6720 <m-t> { encoding = OT1 }
6721 <cmr> { encoding = {OT1,OT4},
6722 <pmn> { encoding = OT1,
6723 <cmr> family = cmr }
6724 <pmn> family = pmnj }
6725 <ebg> { }
6726 {
6727 <m-t|cmr> \AE = {50, },
6728 <pmn> \OE = {50, }
6729 <*cmr|ebg>
6730 "00 = { ,150}, % \Gamma
6731 "01 = {100,100}, % \Delta
6732 "02 = { 50, 50}, % \Theta
6733 "03 = {100,100}, % \Lambda
6734 <ebg> "04 = { 50, 50}, % \Xi
6735 <cmr> "06 = { 50, 50}, % \Sigma
6736 "07 = {100,100}, % \Upsilon
6737 "08 = { 50, 50}, % \Phi
6738 "09 = { 50, 50}, % \Psi
6739 <ebg> "0A = { 50, 50}, % \Omega
6740 <ebg> 138 = { , 50}, % \L

```

Remaining slots can be found in the source file.

```
6741 </cmr|ebg>
6742   }
6743
```

Settings for figure variants.

```
6744 <*ebg>
6745 \SetProtrusion
6746   [ name      = EBGaramond-OT1-LF,
6747     load      = EBGaramond-OT1 ]
6748   { encoding = OT1,
6749     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
6750   {
6751     1 = {50,50},
6752     2 = {50,50},
6753     4 = {50,50},
6754     7 = {50,50},
6755   }
6756
6757 \SetProtrusion
6758   [ name      = EBGaramond-OT1-T0sF,
6759     load      = EBGaramond-OT1 ]
6760   { encoding = OT1,
6761     family   = {EBGaramond-T0sF} }
6762   {
6763     1 = {150,150},
6764     2 = {50,50},
6765     3 = {50,50},
6766     4 = {50,50},
6767     5 = {50,50},
6768     6 = {50,50},
6769     7 = {50,80},
6770     8 = {50,50},
6771     9 = {50,50},
6772   }
6773
6774 </ebg>
6775 </m-t|cmr|ebg|pmn>
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For X_YTeX (EU1) and LuaTeX (EU2) we simply use the T1 list as default (for now).

```
6776 \SetProtrusion
6777 <m-t> [ name      = T1-default,
6778 <bch> [ name      = bch-T1,
6779 <blg> [ name      = blg-T1,
6780 <cmr> [ name      = cmr-T1,
6781 <ebg> [ name      = EBGaramond-T1,
6782 <pmn> [ name      = pmnj-T1,
6783 <ppl> [ name      = ppl-T1,
6784 <ptm> [ name      = ptm-T1,
6785 <ugm> [ name      = ugm-T1,
6786 <m-t>   load      = default ]
6787 <bch>   load      = bch-default ]
6788 <blg>   load      = blg-default ]
6789 <cmr>   load      = cmr-default ]
6790 <ebg>   load      = EBGaramond-default ]
6791 <pmn>   load      = pmnj-default ]
6792 <ppl>   load      = ppl-default ]
6793 <ptm>   load      = ptm-default ]
6794 <ugm>   load      = ugm-default ]
6795 <m-t>   { encoding = {T1,LY1,EU1,EU2,TU} }
6796 <bch|cmr|pmn|ppl> { encoding = {T1,LY1},
6797 <blg|ptm|ugm>   { encoding = {T1},
```

```

6798 <ebg> { encoding = {LY1},
6799 <bch> family = bch }
6800 <blg> family = blg }
6801 <cmr> family = cmr }
6802 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-0sF,EBGaramond-T0sF} }
6803 <pmn> family = pmnj }
6804 <ppl> family = {ppl,pplx,pplj} }
6805 <ptm> family = {ptm,ptmx,ptmj} }
6806 <ugm> family = ugm }
6807 {
6808 <m-t|cmr> \AE = {50, },
6809 <bch|pmn> \OE = {50, },
6810 <pmn> \TH = { ,50},
6811 <blg> \v L = { ,250},
6812 <blg> \v d = { ,250},
6813 <blg> \v l = { ,250},
6814 <blg> \v t = { ,250},
6815 <blg> 127 = {300,400},
6816 <blg> 156 = {100, }, % IJ
6817 <blg> 188 = { 80, 80}, % ij
6818 <m-t|bch|ebg|pmn|ppl|ptm> _ = {100,100},
6819 <cmr> _ = {200,200},
6820 <ugm> _ = {100,200},
6821 <m-t|ebg|pmn|ptm> \textbackslash = {100,200},
6822 <bch> \textbackslash = {150,200},
6823 <blg> \textbackslash = {250,300},
6824 <cmr|ppl> \textbackslash = {200,300},
6825 <ugm> \textbackslash = {100,300},
6826 <ugm> \textbar = {200,200},
6827 <blg> \textendash = {300,300}, \textemdash = {150,150},
6828 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
6829 <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.

```

6830 <m-t|cmr|ebg|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
6831 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
6832 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
6833 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
6834 <blg> \guilsinglleft = {300,500}, \guilsingright = {300,500},
6835 <cmr|ebg|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
6836 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
6837 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
6838 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
6839 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
6840 <blg|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
6841 <ebg> \guillemotleft = {300,300}, \guillemotright = {200,300},
6842 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
6843 <m-t|bch|cmr|ebg|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
6844 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
6845 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
6846 <m-t|cmr|ebg|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
6847 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
6848 <m-t|bch|cmr|ebg|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
6849 <pmn> \textless = {100, }, \textgreater = { ,100},
6850 <pmn> \textvisiblespace = {100,100} % not in LY1
6851 }
6852

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

6853 <*cmr>
6854 \SetProtrusion
6855 [ name = lmr-T1,

```

```

6856     load    = cmr-T1 ]
6857     { encoding = {T1,LY1},
6858       family   = lmr      }
6859     {
6860       \textquotedblleft = {300,400}, \textquotedblright = {300,400}
6861     }
6862
6863 </cmr>
6864 <*ebg>
6865 \SetProtrusion
6866 [ name    = EBGaramond-T1-LF,
6867   load    = EBGaramond-T1 ]
6868 { encoding = T1,
6869   family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
6870 {
6871   1 = {50,50},
6872   2 = {50,50},
6873   4 = {50,50},
6874   7 = {50,50},
6875 }
6876
6877 \SetProtrusion
6878 [ name    = EBGaramond-T1-T0sF,
6879   load    = EBGaramond-T1 ]
6880 { encoding = T1,
6881   family   = {EBGaramond-T0sF} }
6882 {
6883   1 = {150,150},
6884   2 = {50,50},
6885   3 = {50,50},
6886   4 = {50,50},
6887   5 = {50,50},
6888   6 = {50,50},
6889   7 = {50,80},
6890   8 = {50,50},
6891   9 = {50,50},
6892 }
6893
6894 </ebg>

```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion). 14

```

6895 <*m-t|cmr|pmn>
6896 \SetProtrusion
6897 <m-t> [ name    = T2A-default,
6898 <cmr> [ name    = cmr-T2A,
6899 <pmn> [ name    = pmnj-T2A,
6900 <m-t>   load    = default   ]
6901 <cmr>   load    = cmr-default ]
6902 <pmn>   load    = pmnj-default ]
6903 { encoding = T2A,
6904 <m-t> }
6905 <cmr>   family   = cmr }
6906 <pmn>   family   = pmnj }
6907 {
6908   \CYRA = {50,50},
6909   \CYRG = { ,50},
6910   \CYRK = { ,50},
6911   \CYRT = {50,50},
6912   \CYRH = {50,50},
6913   \CYRU = {50,50},
6914 <pmn>   \CYRS = {50, },
6915 <pmn>   \CYRO = {50,50},
6916   \cyrk = { ,50},
6917   \cyrg = { ,50},

```

```

6918     \cyrh = {50,50},
6919 <m-t|pmn> \cyru = {50,50},
6920 <cmr>     \cyru = {50,70},
6921 <m-t>     - = {100,100},
6922 <cmr>     - = {200,200},
6923 <m-t>     \textbackslash = {100,200}, \quotedblbase = {400,400},
6924 <cmr>     \textbackslash = {200,300}, \quotedblbase = {400,400},
6925 <pmn>     \textbackslash = {100,200}, \quotedblbase = {300,300},
6926 <cmr>     \textquotedbl = {300,300}, \textquotedblleft = {200,600},
6927 <m-t>     \guillemotleft = {200,200}, \guillemotright = {200,200},
6928 <cmr>     \guillemotleft = {300,200}, \guillemotright = {100,400},
6929 <pmn>     \guillemotleft = {200,200}, \guillemotright = {150,300},
6930 <m-t|cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
6931 <pmn>     \textbraceleft = {200, }, \textbraceright = { ,300},
6932 <m-t|cmr> \textless = {200,100}, \textgreater = {100,200}
6933 <pmn>     \textless = {100, }, \textgreater = { ,100}
6934 }
6935
6936 </m-t|cmr|pmn>

```

Settings for the QX encoding (generic and Times).¹⁵ It also includes some glyphs otherwise in TS1.

```

6937 <*m-t|ptm>
6938 \SetProtrusion
6939 <m-t> [ name = QX-default,
6940 <ptm> [ name = ptm-QX,
6941 <m-t> load = default ]
6942 <ptm> load = ptm-default ]
6943 <m-t> { encoding = QX }
6944 <ptm> { encoding = QX,
6945 <ptm> family = {ptm,ptmx,ptmj} }
6946 {
6947     \AE = {50, },
6948 <ptm> * = {200,200},
6949     {=} = {100,100},
6950     \textunderscore = {100,100},
6951     \textbackslash = {100,200},
6952     \quotedblbase = {400,400},
6953 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
6954 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
6955     \textexclamdown = {100, }, \textquestiondown = {100, },
6956 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
6957 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
6958     \textless = {200,100}, \textgreater = {100,200},
6959     \textminus = {200,200}, \textdegree = {300,300},
6960 <m-t> \copyright = {100,100}, \textregistered = {100,100}
6961 <ptm> \copyright = {100,150}, \textregistered = {100,150},
6962 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
6963 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
6964 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
6965 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
6966 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
6967 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
6968 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
6969 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
6970 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
6971 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
6972 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
6973 <ptm> \textperthousand = { ,50}
6974 }
6975
6976 </m-t|ptm>

```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented

15 Contributed by Maciej Eder.

characters are already taken care of by the inheritance list.

```

6977 <*cmr|bch>
6978 \SetProtrusion
6979 <cmr> [ name = cmr-T5,
6980 <cmr> load = cmr-default ]
6981 <bch> [ name = bch-T5,
6982 <bch> load = bch-default ]
6983 { encoding = T5,
6984 <cmr> family = cmr }
6985 <bch> family = bch }
6986 {
6987 <bch> _ = {100,100},
6988 <bch> \textbackslash = {150,200},
6989 <cmr> \textbackslash = {200,300},
6990 <cmr> \textquotedblleft = {200,600},
6991 <cmr> \textquotedbl = {300,300},
6992 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
6993 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
6994 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
6995 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
6996 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
6997 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
6998 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
6999 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
7000 \textless = {200,100}, \textgreater = {100,200}
7001 }
7002
7003 </cmr|bch>

```

Minion with lining numbers.

```

7004 <*pmn>
7005 \SetProtrusion
7006 [ name = pmnx-OT1,
7007 load = pmnj-default ]
7008 { encoding = OT1,
7009 family = pmnx }
7010 {
7011 1 = {230,180}
7012 }
7013
7014 \SetProtrusion
7015 [ name = pmnx-T1,
7016 load = pmnj-T1 ]
7017 { encoding = {T1,LY1},
7018 family = pmnx }
7019 {
7020 1 = {230,180}
7021 }
7022
7023 \SetProtrusion
7024 [ name = pmnx-T2A,
7025 load = pmnj-T2A ]
7026 { encoding = {T2A},
7027 family = pmnx }
7028 {
7029 1 = {230,180}
7030 }
7031
7032 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

7033 <*ptm>
7034 \SetProtrusion
7035 [ name = ptm-LY1,

```

```

7036   load   = ptm-T1 ]
7037   { encoding = LY1,
7038     family   = {ptm,ptmx,ptmj} }
7039   {
7040     -                = {100,100},
7041     \texttrademark  = {100,100},
7042     \textregistered = {100,100},
7043     \textcopyright  = {100,100},
7044     \textdegree     = {300,300},
7045     \textminus      = {200,200},
7046     \textellipsis   = {150,200},
7047 % \texteuro        = { , }, % ?
7048     \textcent       = {100,100},
7049     \textquotesingle = {500,500},
7050     \textflorin     = { 50, 70},
7051     \textdagger     = {150,150},
7052     \textdaggerdbl  = {100,100},
7053     \textperthousand = { , 50},
7054     \textbullet     = {150,150},
7055     \textonesuperior = {100,100},
7056     \texttwosuperior = { 50, 50},
7057     \textthreesuperior = { 50, 50},
7058     \textperiodcentered = {300,300},
7059     \textplusminus  = { 50, 80},
7060     \textmultiply   = {100,100},
7061     \textdivide     = { 50,150}

```

Remaining slots in the source file.

```

7062   }
7063
7064 </ptm>

```

For the Greek LGR encoding.

```

7065 <*ebg>
7066 \SetProtrusion
7067 [ name = EBGaramond-LGR ]
7068 { }
7069 {
7070   A = {50,50},
7071   D = {100,100},
7072   F = {50,50},
7073   G = { ,150},
7074   K = { ,50},
7075   L = {100,100},
7076   O = {50,50},
7077   U = {100,100},
7078   T = {50,50},
7079   W = { ,50},
7080   Y = {50,50},
7081   . = { ,600},
7082   {,} = { ,500},
7083   : = { ,400},
7084   ; = { ,300},
7085   ! = { ,100},
7086   ? = { ,100},
7087   ~ = {200,250},
7088   \% = {50,50},
7089   * = {300,300},
7090   + = {250,250},
7091   {=} = { 50, 50},
7092   ( = {100, }, ) = { ,200},
7093   / = {100,200},
7094   - = {300,500},
7095   \texteuro = { 50,100},
7096   \textendash = {300,300}, \textemdash = {200,200},

```

```

7097 \textquoteleft = {300,500}, \textquoteright = {400,400},
7098 \guillemotleft = {300,300}, \guillemotright = {200,400},
7099 }
7100
7101 \SetProtrusion
7102 [ name = EBGaramond-LGR-LF,
7103 load = EBGaramond-LGR ]
7104 { encoding = LGR,
7105 family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF} }
7106 {
7107 1 = {50,50},
7108 2 = {50,50},
7109 4 = {50,50},
7110 7 = {50,50},
7111 }
7112
7113 \SetProtrusion
7114 [ name = EBGaramond-LGR-TOfF,
7115 load = EBGaramond-LGR ]
7116 { encoding = LGR,
7117 family = {EBGaramond-TOfF} }
7118 {
7119 1 = {150,150},
7120 2 = {50,50},
7121 3 = {50,50},
7122 4 = {50,50},
7123 5 = {50,50},
7124 6 = {50,50},
7125 7 = {50,80},
7126 8 = {50,50},
7127 9 = {50,50},
7128 }
7129
7130 </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.¹⁶

```

7131 \SetProtrusion
7132 <m-t> [ name = OT1-it ]
7133 <bch> [ name = bch-it ]
7134 <blg> [ name = blg-it,
7135 <blg> load = blg-default ]
7136 <cmr> [ name = cmr-it ]
7137 <ebg> [ name = EBGaramond-it ]
7138 <pmn> [ name = pmn-it ]
7139 <ppl> [ name = ppl-it ]
7140 <ptm> [ name = ptm-it ]
7141 <ugm> [ name = ugm-it ]
7142 <m-t|bch|blg|ugm> { encoding = OT1,
7143 <ppl|ptm> { encoding = {OT1,OT4},
7144 <bch> family = bch,
7145 <blg> family = blg,
7146 <ppl> family = {ppl,pplx,pplj},
7147 <ptm> family = {ptm,ptmx,ptmj},

```

¹⁶ Settings contributed by *Hendrik Vogt*.


```

7148 <ugm>      family = ugm,
7149 <m-t|bch|ppl|ptm>  shape = {it,s1} }
7150 <blg|ugm>    shape = it }
7151 <cmr|ebg|pmn>  { }
7152 {
7153 <cmr>      A = {100,100},
7154 <ptm>      A = {100,50},
7155 <ebg|pmn>   A = {50, },
7156 <ugm>      A = { ,150},
7157 <ppl>      A = {50,50},
7158 <ptm>      \AE = {100, },
7159 <ebg|ppl>   \AE = {50, },
7160 <cmr>      B = {83,-40},
7161 <ebg|ppl|ptm>  B = {50, },
7162 <pmn>      B = {20,-50},
7163 <bch|ppl|ptm|ugm> C = {50, },
7164 <cmr>      C = {165,-75},
7165 <ebg>      C = {100, },
7166 <pmn>      C = {50,-50},
7167 <cmr>      D = {75, -28},
7168 <ebg|ppl|ptm> D = {50,50},
7169 <pmn>      D = {20, },
7170 <cmr>      E = {80,-55},
7171 <ebg|ppl|ptm> E = {50, },
7172 <pmn>      E = {20,-50},
7173 <cmr>      F = {85,-80},
7174 <ebg|ptm>  F = {100, },
7175 <pmn>      F = {10, },
7176 <ppl>      F = {50, },
7177 <bch|ppl|ptm|ugm> G = {50, },
7178 <cmr>      G = {153,-15},
7179 <ebg>      G = {100, },
7180 <pmn>      G = {50,-50},
7181 <cmr>      H = {73,-60},
7182 <ebg|ppl|ptm> H = {50, },
7183 <cmr>      I = {140,-120},
7184 <ebg|ptm>  I = {50, },
7185 <pmn>      I = {20,-50},
7186 <cmr>      J = {135,-80},
7187 <ebg>      J = {50, },
7188 <pmn>      J = {20, },
7189 <ptm>      J = {100, },
7190 <cmr>      K = {70,-30},
7191 <ebg|ppl|ptm> K = {50, },
7192 <pmn>      K = {20, },
7193 <cmr>      L = {87, 40},
7194 <ebg|ppl|ptm> L = {50, },
7195 <pmn>      L = {20,50},
7196 <ugm>      L = { ,100},
7197 <cmr>      M = {67,-45},
7198 <pmn>      M = { , -30},
7199 <ptm>      M = {50, },
7200 <cmr>      N = {75,-55},
7201 <pmn>      N = { , -30},
7202 <ptm>      N = {50, },
7203 <bch|pmn|ppl|ptm> O = {50, },
7204 <cmr>      O = {150,-30},
7205 <ebg>      O = {100, },
7206 <ugm>      O = {70,50},
7207 <ppl|ptm>  \OE = {50, },
7208 <ebg>      \OE = {100, },
7209 <cmr>      P = {82,-50},
7210 <ebg|ppl|ptm> P = {50, },
7211 <pmn>      P = {20,-50},
7212 <bch|pmn|ppl|ptm> Q = {50, },

```

7213 <cmr> Q = {150,-30},
7214 <ebg> Q = {100, },
7215 <ugm> Q = {70,50},
7216 <cmr> R = {75, 15},
7217 <ebg|ppl|ptm> R = {50, },
7218 <pmn> R = {20, },
7219 <bch|ebg|ppl|ptm> S = {50, },
7220 <cmr> S = {90,-65},
7221 <pmn> S = {20,-30},
7222 <bch|ebg|ppl|ptm> \$ = {50, },
7223 <cmr> \$ = {100,-20},
7224 <pmn> \$ = {20,-30},
7225 <bch|pmn|ugm> T = {70, },
7226 <cmr> T = {220,-85},
7227 <ebg|ppl|ptm> T = {100, },
7228 <cmr> U = {230,-55},
7229 <ebg|ppl|ptm> U = {50, },
7230 <pmn> U = {50,-50},
7231 <cmr> V = {260,-60},
7232 <ebg|pmn|ugm> V = {100, },
7233 <ppl|ptm> V = {100,50},
7234 <cmr> W = {185,-55},
7235 <ebg|pmn|ugm> W = {100, },
7236 <ppl> W = {50, },
7237 <ptm> W = {100,50},
7238 <cmr> X = {70,-30},
7239 <ppl|ptm> X = {50, },
7240 <cmr> Y = {250,-60},
7241 <pmn> Y = {50, },
7242 <ppl> Y = {100,50},
7243 <ptm> Y = {100, },
7244 <cmr> Z = {90,-60},
7245 <pmn> Z = { , -50},
7246 <cmr> a = {150,-10},
7247 <cmr> b = {170, },
7248 <cmr> c = {173,-10},
7249 <cmr> d = {150,-55},
7250 <pmn> d = { , -50},
7251 <cmr> e = {180, },
7252 <cmr> f = { , -250},
7253 <ebg|pmn> f = { , -100},
7254 <cmr> g = {150,-10},
7255 <cmr> h = {100, },
7256 <cmr> i = {210, },
7257 <pmn> i = { , -30},
7258 <cmr> j = { , -40},
7259 <pmn> j = { , -30},
7260 <cmr> k = {110,-50},
7261 <cmr> l = {240,-110},
7262 <pmn> l = { , -100},
7263 <cmr> m = {80, },
7264 <cmr> n = {115, },
7265 <bch> o = {50,50},
7266 <cmr> o = {155, },
7267 <bch> p = { , 50},
7268 <pmn> p = {-50, },
7269 <bch> q = {50, },
7270 <cmr> q = {170,-40},
7271 <cmr> r = {155,-40},
7272 <pmn> r = { , 50},
7273 <cmr> s = {130, },
7274 <bch> t = { , 50},
7275 <cmr> t = {230,-10},
7276 <cmr> u = {120, },
7277 <cmr> v = {140,-25},

```

7278 <pmn|ugm>    v = {50, },
7279 <bch>        w = { ,50},
7280 <cmr>        w = {98,-20},
7281 <pmn|ugm>    w = {50, },
7282 <cmr>        x = {65,-40},
7283 <bch>        y = { ,50},
7284 <cmr>        y = {130,-20},
7285 <cmr>        z = {110,-80},
7286 <cmr>        0 = {170,-85},
7287 <bch|ptm>    1 = {150,100},
7288 <cmr>        1 = {230,110},
7289 <ebg>        1 = {150, },
7290 <pmn>        1 = {50, },
7291 <ppl>        1 = {100, },
7292 <ugm>        1 = {150,150},
7293 <cmr>        2 = {130,-70},
7294 <ebg|ppl|ptm> 2 = {50, },
7295 <pmn>        2 = {-50, },
7296 <bch>        3 = {50, },
7297 <cmr>        3 = {140,-70},
7298 <pmn>        3 = {-100, },
7299 <ptm>        3 = {100,50},
7300 <bch>        4 = {100, },
7301 <cmr>        4 = {130,80},
7302 <ebg>        4 = {150, },
7303 <ppl|ptm>    4 = {50, },
7304 <cmr>        5 = {160, },
7305 <ptm>        5 = {50, },
7306 <bch>        6 = {50, },
7307 <cmr>        6 = {175,-30},
7308 <bch|ebg|ptm> 7 = {100, },
7309 <cmr>        7 = {250,-150},
7310 <pmn>        7 = {20, },
7311 <ppl>        7 = {50, },
7312 <cmr>        8 = {130,-40},
7313 <cmr>        9 = {155,-80},
7314 <m-t|cmr|ebg|pmn|ppl> . = { ,500},
7315 <blg>        . = {400,600},
7316 <bch|ptm|ugm> . = { ,700},
7317 <blg>        {,}= {300,500},
7318 <m-t|ebg|pmn|ppl> {,}= { ,500},
7319 <cmr>        {,}= { ,450},
7320 <bch|ugm>    {,}= { ,600},
7321 <ptm>        {,}= { ,700},
7322 <m-t|cmr|ebg|ppl> := { ,300},
7323 <bch|ugm>    := { ,400},
7324 <pmn>        := { ,200},
7325 <ptm>        := { ,500},
7326 <m-t|cmr|ebg|ppl> ; = { ,300},
7327 <bch|ugm>    ; = { ,400},
7328 <pmn>        ; = { ,200},
7329 <ptm>        ; = { ,500},
7330 <ptm>        ! = { ,100},
7331 <bch>        ? = { ,200},
7332 <ptm>        ? = { ,100},
7333 <ppl>        ? = { ,300},
7334 <pmn>        " = {400,200},
7335 <m-t|ebg|pmn|ppl|ptm> & = {50,50},
7336 <bch>        & = { ,80},
7337 <cmr>        & = {130,30},
7338 <ugm>        & = {50,100},
7339 <m-t|ebg|pmn> \% = {100, },
7340 <cmr>        \% = {180,50},
7341 <bch>        \% = {50,50},
7342 <ppl|ptm>    \% = {100,100},

```

```

7343 <ugm> \% = {100,50},
7344 <m-t|pmn|ppl> * = {200,200},
7345 <bch> * = {300,200},
7346 <cmr> * = {380,20},
7347 <ebg> * = {500,100},
7348 <ptm|ugm> * = {400,200},
7349 <m-t|pmn|ppl> + = {150,200},
7350 <cmr> + = {180,200},
7351 <bch|ugm> + = {250,250},
7352 <ebg|ptm> + = {250,200},
7353 <m-t|ebg|pmn|ppl> @ = {50,50},
7354 <bch> @ = {80,50},
7355 <cmr> @ = {180,10},
7356 <ptm> @ = {150,150},
7357 <m-t|bch|ugm> ~ = {150,150},
7358 <cmr|ebg|pmn|ppl|ptm> ~ = {200,150},
7359 <ugm> {=} = {200,200},
7360 <m-t|bch|ebg|pmn|ppl|ptm|ugm> ( = {200, }, ) = { ,200},
7361 <cmr> ( = {300, }, ) = { ,70},
7362 <m-t|ebg|ppl|ptm|ugm> / = {100,200},
7363 <cmr> / = {100,100},
7364 <bch> / = { ,150},
7365 <pmn> / = {100,150},
7366 <m-t> - = {300,300},
7367 <bch|ebg> - = {300,400},
7368 <pmn> - = {200,300},
7369 <cmr> - = {500,300},
7370 <ppl> - = {300,500},
7371 <ptm> - = {500,500},
7372 <ugm> - = {400,700},
7373 <blg> - = {0,300},
7374 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
7375 <bch> \textendash = {200,300}, \textendash = {150,200},
7376 <cmr> \textendash = {500,300}, \textendash = {400,170},
7377 <ebg|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
7378 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
7379 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
7380 <cmr> \textquoteleft = {800,200}, \textquoteright = {800,-20},
7381 <ebg> \textquoteleft = {800,200}, \textquoteright = {800,200},
7382 <ppl> \textquoteleft = {700,400}, \textquoteright = {700,400},
7383 <ptm> \textquoteleft = {800,500}, \textquoteright = {800,500},
7384 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
7385 <blg> \textquotedblright = {300,300}
7386 <cmr> \textquotedblleft = {540,100}, \textquotedblright = {500,100}
7387 <ebg> \textquotedblleft = {700,200}, \textquotedblright = {700,200}
7388 <ppl> \textquotedblleft = {500,300}, \textquotedblright = {500,300}
7389 <ptm> \textquotedblleft = {700,400}, \textquotedblright = {700,400}
7390 <ugm> \textquotedblleft = {600,200}, \textquotedblright = {600,200}
7391 }
7392
7393 <*cmr|ebg|pmn>
7394 \SetProtrusion
7395 <cmr> [ name = cmr-it-OT1,
7396 <ebg> [ name = EBGaramond-it-OT1,
7397 <pmn> [ name = pmnj-it-OT1,
7398 <cmr> load = cmr-it ]
7399 <ebg> load = EBGaramond-it ]
7400 <pmn> load = pmnj-it ]
7401 <cmr> { encoding = {OT1,OT4},
7402 <pmn> { encoding = OT1,
7403 <cmr> family = cmr,
7404 <pmn> family = pmnj,
7405 <cmr> shape = it }
7406 <pmn> shape = {it,s1} }
7407 <ebg> { }

```

```

7408 {
7409 <cmr> \AE = {100, },
7410 <pmn> \AE = { , -50},
7411 <cmr> \OE = {100, },
7412 <pmn> \OE = {50, }
7413 <*cmr|ebg>
7414 <cmr> "00 = {200,150}, % \Gamma
7415 <ebg> "00 = { ,150}, % \Gamma
7416 <cmr> "01 = {150,100}, % \Delta
7417 <ebg> "01 = {100,100}, % \Delta
7418 <cmr> "02 = {150, 50}, % \Theta
7419 <ebg> "02 = { 50, 50}, % \Theta
7420 <cmr> "03 = {150, 50}, % \Lambda
7421 <ebg> "03 = {100,100}, % \Lambda
7422 <cmr> "04 = {100,100}, % \Xi
7423 <ebg> "04 = { 50, 50}, % \Xi
7424 <cmr> "05 = {100,100}, % \Pi
7425 <cmr> "06 = {100, 50}, % \Sigma
7426 <cmr> "07 = {200,150}, % \Upsilon
7427 <ebg> "07 = {100,100}, % \Upsilon
7428 <cmr> "08 = {150, 50}, % \Phi
7429 <ebg> "08 = { 50, 50}, % \Phi
7430 <cmr> "09 = {150,100}, % \Psi
7431 <ebg> "09 = { 50, 50}, % \Psi
7432 "0A = { 50, 50}, % \Omega
7433 <ebg> 138 = { , 50}, % \L
7434 </cmr|ebg>
7435 }
7436
7437 </cmr|ebg|pmn>
7438 <*ebg>
7439 \SetProtrusion
7440 [ name = EBGaramond-it-OT1-LF,
7441 load = EBGaramond-it-OT1 ]
7442 { encoding = OT1,
7443 family = {EBGaramond-LF,EBGaramond-TLF},
7444 shape = it }
7445 {
7446 1 = {50,50},
7447 2 = {50,50},
7448 3 = {80,50},
7449 4 = {50,50},
7450 5 = {50,50},
7451 6 = {50,50},
7452 7 = {50,50},
7453 8 = {50,50},
7454 9 = {50, },
7455 }
7456
7457 \SetProtrusion
7458 [ name = EBGaramond-it-OT1-0sF,
7459 load = EBGaramond-it-OT1 ]
7460 { encoding = OT1,
7461 family = {EBGaramond-0sF},
7462 shape = it }
7463 {
7464 1 = {50,50},
7465 2 = {50,50},
7466 3 = { ,80},
7467 4 = {50,50},
7468 7 = {50,50},
7469 }
7470
7471 \SetProtrusion
7472 [ name = EBGaramond-it-OT1-T0sF,

```

```

7473     load      = EBGaramond-it-OT1 ]
7474     { encoding = OT1,
7475       family   = {EBGaramond-T0sF},
7476       shape    = it }
7477     {
7478     0 = {150,150},
7479     1 = {150,150},
7480     2 = {80,80},
7481     3 = {50,80},
7482     4 = {50,80},
7483     5 = {50,80},
7484     6 = {50,50},
7485     7 = {50,100},
7486     8 = {50,50},
7487     9 = {50,80},
7488     }
7489
7490 </ebg>
7491 \SetProtrusion
7492 <m-t> [ name      = T1-it-default,
7493 <bch> [ name      = bch-it-T1,
7494 <blg> [ name      = blg-it-T1,
7495 <cmr> [ name      = cmr-it-T1,
7496 <ebg> [ name      = EBGaramond-it-T1,
7497 <pmn> [ name      = pmnj-it-T1,
7498 <ppl> [ name      = ppl-it-T1,
7499 <ptm> [ name      = ptm-it-T1,
7500 <ugm> [ name      = ugm-it-T1,
7501 <m-t> load      = OT1-it ]
7502 <bch> load      = bch-it ]
7503 <blg> load      = blg-T1 ]
7504 <cmr> load      = cmr-it ]
7505 <pmn> load      = pmnj-it ]
7506 <ebg> load      = EBGaramond-it ]
7507 <ppl> load      = ppl-it ]
7508 <ptm> load      = ptm-it ]
7509 <ugm> load      = ugm-it ]
7510 <m-t|bch|cmr|pmn|ppl> { encoding = {T1,LY1},
7511 <ebg> { encoding = {LY1},
7512 <blg|ptm|ugm> { encoding = T1,
7513 <bch> family   = bch,
7514 <blg> family   = blg,
7515 <cmr> family   = cmr,
7516 <pmn> family   = pmnj,
7517 <ebg> family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-0sF,EBGaramond-T0sF},
7518 <ppl> family   = {ppl,pplx,pplj},
7519 <ptm> family   = {ptm,ptmx,ptmj},
7520 <ugm> family   = ugm,
7521 <m-t|bch|pmn|ppl|ptm> shape = {it,s1} }
7522 <blg|cmr|ebg|ugm> shape = it }
7523 {
7524 <m-t|bch|pmn> _ = { ,100},
7525 <blg> _ = {0,300},
7526 <cmr|ugm> _ = {100,200},
7527 <ebg|ppl|ptm> _ = {100,100},
7528 <blg> . = {400,600},
7529 <blg> {,}= {300,500},
7530 <cmr> \AE = {100, },
7531 <pmn> \AE = { , -50},
7532 <bch|pmn> \OE = { 50, },
7533 <cmr> \OE = {100, },
7534 <pmn> 031 = { , -100}, % ffl
7535 <cmr|ptm> 156 = {100, }, % IJ
7536 <ebg> 156 = {50, }, % IJ
7537 <pmn> 156 = {20, }, % IJ

```

```

7538 <pmn> 188 = { , -30}, % ij
7539 <pmn> \v t = { , 100},
7540 <m-t|ebg|ppl|ptm> \textbackslash = {100,200},
7541 <cmr|ugm> \textbackslash = {300,300},
7542 <bch> \textbackslash = {150,150},
7543 <pmn> \textbackslash = {100,150},
7544 <ugm> \textbar = {200,200},
7545 <cmr> \textquotedblleft = {500,300},
7546 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
7547 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
7548 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
7549 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
7550 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
7551 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
7552 <ebg|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
7553 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
7554 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
7555 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
7556 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
7557 <ebg> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
7558 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
7559 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
7560 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
7561 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
7562 <ebg> \guillemotleft = {300,300}, \guillemotright = {200,400},
7563 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
7564 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
7565 <m-t|ebg|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
7566 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
7567 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
7568 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
7569 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
7570 <cmr|ebg|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
7571 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
7572 <cmr|ebg|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
7573 <pmn> \textvisiblespace = {100,100}
7574 }
7575
7576 <*ebg>
7577 \SetProtrusion
7578 [ name = EBGaramond-it-T1-LF,
7579 load = EBGaramond-it-T1 ]
7580 { encoding = T1,
7581 family = {EBGaramond-LF,EBGaramond-TLF},
7582 shape = it }
7583 {
7584 1 = {50,50},
7585 2 = {50,50},
7586 3 = {80,50},
7587 4 = {50,50},
7588 5 = {50,50},
7589 6 = {50,50},
7590 7 = {50,50},
7591 8 = {50,50},
7592 9 = {50, },
7593 }
7594
7595 \SetProtrusion
7596 [ name = EBGaramond-it-T1-OsF,
7597 load = EBGaramond-it-T1 ]
7598 { encoding = T1,
7599 family = {EBGaramond-OsF},
7600 shape = it }
7601 {
7602 1 = {50,50},

```

```

7603     2 = {50,50},
7604     3 = { ,80},
7605     4 = {50,50},
7606     7 = {50,50},
7607   }
7608
7609 \SetProtrusion
7610   [ name      = EBGaramond-it-T1-T0sF,
7611     load      = EBGaramond-it-T1 ]
7612   { encoding = T1,
7613     family   = {EBGaramond-T0sF},
7614     shape    = it }
7615   {
7616     0 = {150,150},
7617     1 = {150,150},
7618     2 = {80,80},
7619     3 = {50,80},
7620     4 = {50,80},
7621     5 = {50,80},
7622     6 = {50,50},
7623     7 = {50,100},
7624     8 = {50,50},
7625     9 = {50,80},
7626   }
7627
7628 (/ebg)
7629 <*m-t|cmr|pmn)
7630 \SetProtrusion
7631 <m-t) [ name      = T2A-it-default,
7632 <cmr) [ name      = cmr-it-T2A,
7633 <pmn) [ name      = pmnj-it-T2A,
7634 <m-t)   load      = OT1-it   ]
7635 <cmr)   load      = cmr-it   ]
7636 <pmn)   load      = pmnj-it   ]
7637   { encoding = T2A,
7638 <cmr)   family   = cmr,
7639 <pmn)   family   = pmnj,
7640 <m-t|pmn) shape   = {it,sl} }
7641 <cmr)   shape   = it       }
7642   {
7643 <cmr)   \CYRA = {100,50},
7644 <pmn)   \CYRA = {50, },
7645 <cmr)   \CYRB = {50, },
7646 <cmr)   \CYRV = {50, },
7647 <pmn)   \CYRV = {20,-50},
7648 <cmr)   \CYRG = {100, },
7649 <pmn)   \CYRG = {10, },
7650 <cmr)   \CYRD = {50, },
7651 <cmr)   \CYRE = {50, },
7652 <pmn)   \CYRE = {20,-50},
7653 <cmr)   \CYRZH = {50, },
7654 <cmr)   \CYRZ = {50, },
7655 <pmn)   \CYRZ = {20,-50},
7656 <cmr)   \CYRI = {50, },
7657 <pmn)   \CYRI = { , -30},
7658 <cmr)   \CYRISHRT = {50, },
7659 <cmr)   \CYRK = {50, },
7660 <pmn)   \CYRK = {20, },
7661 <cmr)   \CYRL = {50, },
7662 <cmr)   \CYRM = {50, },
7663 <pmn)   \CYRM = { , -30},
7664 <cmr)   \CYRN = {50, },
7665 <cmr)   \CYRO = {100, },
7666 <pmn)   \CYRO = {50, },
7667 <cmr)   \CYRP = {50, },

```



```

7668 <cmr> \CYRR = {50, },
7669 <pmn> \CYRR = {20,-50},
7670 <cmr> \CYRS = {100, },
7671 <pmn> \CYRS = {50, },
7672 <cmr> \CYRT = {100, },
7673 <pmn> \CYRT = {70, },
7674 <cmr> \CYRU = {100, },
7675 <pmn> \CYRU = {50, },
7676 <cmr> \CYRF = {100, },
7677 <cmr> \CYRH = {50, },
7678 <cmr> \CYRC = {50, },
7679 <cmr> \CYRCH = {100, },
7680 <cmr> \CYRSH = {50, },
7681 <cmr> \CYRSHCH = {50, },
7682 <cmr> \CYRHRDSN = {100, },
7683 <cmr> \CYRERY = {50, },
7684 <cmr> \CYRSFTSN = {50, },
7685 <cmr> \CYREREV = {50, },
7686 <cmr> \CYRYU = {50, },
7687 <cmr> \CYRYA = {50, },
7688 <pmn> \CYRYA = { ,20},
7689 <pmn> \cyrr = {-50, },
7690 <m-t|pmn> _ = { ,100},
7691 <cmr> _ = {100,200},
7692 <pmn> 031 = { ,-100}, % ff1
7693 <pmn> \v t = { ,100},
7694 <m-t> \textbackslash = {100,200}, \quotedblbase = {400,500},
7695 <cmr> \textbackslash = {300,300}, \quotedblbase = {200,600},
7696 <pmn> \textbackslash = {100,150}, \quotedblbase = {150,500},
7697 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
7698 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
7699 <pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
7700 <m-t> \textbraceleft = {200,100}, \textbraceright = {200,200},
7701 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
7702 <pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
7703 <cmr> \textquotedblleft = {500,300},
7704 <cmr> \textless = {300,100}, \textgreater = {200,100}
7705 <pmn> \textless = {100, }, \textgreater = { ,100}
7706 }
7707
7708 </m-t|cmr|pmn>
7709 <*m-t|ptm>
7710 \SetProtrusion
7711 <m-t> [ name = QX-it-default,
7712 <ptm> [ name = ptm-it-QX,
7713 <m-t> load = OT1-it ]
7714 <ptm> load = ptm-it ]
7715 { encoding = {QX},
7716 <ptm> family = {ptm,ptmx,ptmj},
7717 shape = {it,sl} }
7718 {
7719 <ptm> 009 = { , 50}, % fk
7720 {=} = {100,100},
7721 <m-t> \textunderscore = {100,100},
7722 <ptm> \textunderscore = {100,150},
7723 \textbackslash = {100,200},
7724 \quotedblbase = {300,400},
7725 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
7726 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
7727 \textexclamdown = {200, }, \textquestiondown = {200, },
7728 \textbraceleft = {200,100}, \textbraceright = {200,200},
7729 \textless = {100,100}, \textgreater = {100,100},
7730 \textminus = {200,200}, \textdegree = {300,150},
7731 <m-t> \copyright = {100,100}, \textregistered = {100,100}
7732 <ptm> \textregistered = {100,150}, \copyright = {100,150},

```

```

7733 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
7734 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
7735 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
7736 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
7737 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
7738 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
7739 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
7740 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
7741 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
7742 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
7743 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
7744 <ptm> \textperthousand = { ,50}
7745 }
7746
7747 </m-t|ptm>
7748 <*cmr|bch>
7749 \SetProtrusion
7750 <cmr> [ name = cmr-it-T5,
7751 <cmr> load = cmr-it ]
7752 <bch> [ name = bch-it-T5,
7753 <bch> load = bch-it ]
7754 { encoding = T5,
7755 <bch> family = bch,
7756 <cmr> family = cmr,
7757 shape = it }
7758 {
7759 <bch> _ = { ,100},
7760 <cmr> _ = {100,200},
7761 <bch> \textbackslash = {150,150},
7762 <cmr> \textbackslash = {300,300},
7763 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
7764 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
7765 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
7766 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
7767 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
7768 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
7769 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
7770 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
7771 <bch> \textless = {100, }, \textgreater = { ,100}
7772 <cmr> \textless = {300,100}, \textgreater = {200,100}
7773 }
7774
7775 </cmr|bch>

```

Slanted is very similar to italic.

```

7776 <*cmr>
7777 \SetProtrusion
7778 [ name = cmr-sl,
7779 load = cmr-it-OT1 ]
7780 { encoding = {OT1,OT4},
7781 family = cmr,
7782 shape = sl }
7783 {
7784 L = { ,50},
7785 f = { , -50},
7786 - = {300, },
7787 \textendash = {400, }, \textemdash = {300, }
7788 }
7789
7790 \SetProtrusion
7791 [ name = cmr-sl-T1,
7792 load = cmr-it-T1 ]
7793 { encoding = {T1,LY1},
7794 family = cmr,
7795 shape = sl }

```

```

7796 {
7797   L = { ,50},
7798   f = { ,-50},
7799   - = {300, },
7800   \textendash = {400, }, \textemdash = {300, }
7801 }
7802
7803 \SetProtrusion
7804 [ name = cmr-sl-T2A,
7805   load = cmr-it-T2A ]
7806 { encoding = T2A,
7807   family = cmr,
7808   shape = sl }
7809 {
7810   L = { ,50},
7811   f = { ,-50},
7812   - = {300, },
7813   \textendash = {400, }, \textemdash = {300, }
7814 }
7815
7816 \SetProtrusion
7817 [ name = cmr-sl-T5,
7818   load = cmr-it-T5 ]
7819 { encoding = T5,
7820   family = cmr,
7821   shape = sl }
7822 {
7823   L = { ,50},
7824   f = { ,-50},
7825   - = {300, },
7826   \textendash = {400, }, \textemdash = {300, }
7827 }
7828
7829 \SetProtrusion
7830 [ name = lmr-it-T1,
7831   load = cmr-it-T1 ]
7832 { encoding = {T1,LY1},
7833   family = lmr,
7834   shape = {it,sl} }
7835 {
7836   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
7837   \quotesinglbase = { ,400}, \quotedblbase = { ,500}
7838 }
7839

```

Oldstyle numerals are slightly different.

```

7840 \SetProtrusion
7841 [ name = cmr(oldstyle)-it,
7842   load = cmr-it-T1 ]
7843 { encoding = T1,
7844   family = {hfor,cmor},
7845   shape = {it,sl} }
7846 {
7847   1 = {250, 50},
7848   2 = {150,-100},
7849   3 = {100,-50},
7850   4 = {150,150},
7851   6 = {200, },
7852   7 = {200, 50},
7853   8 = {150,-50},
7854   9 = {100, 50}
7855 }
7856
7857 </cmr>
7858 <*pmn>

```

```

7859 \SetProtrusion
7860 [ name = pmnx-it,
7861   load = pmnj-it ]
7862 { encoding = OT1,
7863   family = pmnx,
7864   shape = {it,s1} }
7865 {
7866   1 = {100,150}
7867 }
7868
7869 \SetProtrusion
7870 [ name = pmnx-it-T1,
7871   load = pmnj-it-T1 ]
7872 { encoding = {T1,LY1},
7873   family = pmnx,
7874   shape = {it,s1} }
7875 {
7876   1 = {100,150}
7877 }
7878
7879 \SetProtrusion
7880 [ name = pmnx-it-T2A,
7881   load = pmnj-it-T2A ]
7882 { encoding = {T2A},
7883   family = pmnx,
7884   shape = {it,s1} }
7885 {
7886   1 = {100,150}
7887 }
7888
7889 (/pmn)
7890 (*ptm)
7891 \SetProtrusion
7892 [ name = ptm-it-LY1,
7893   load = ptm-it-T1 ]
7894 { encoding = {LY1},
7895   family = {ptm,ptmx,ptmj},
7896   shape = {it,s1} }
7897 {
7898   - = {100,100},
7899   \texttrademark = {100,100},
7900   \textregistered = {100,100},
7901   \textcopyright = {100,100},
7902   \textdegree = {300,100},
7903   \textminus = {200,200},
7904   \textellipsis = {100,200},
7905   \% \texteuro = { , }, % ?
7906   \textcent = {100,100},
7907   \textquotesingle = {500, },
7908   \textflorin = {100, 70},
7909   \textdagger = {150,150},
7910   \textdaggerdbl = {100,100},
7911   \textbullet = {150,150},
7912   \textonesuperior = {150,100},
7913   \texttwosuperior = {150, 50},
7914   \textthreesuperior = {150, 50},
7915   \textparagraph = {100, },
7916   \textperiodcentered = {500,300},
7917   \textonequarter = { 50, },
7918   \textonehalf = { 50, },
7919   \textplusminus = {100,100},
7920   \textmultiply = {150,150},
7921   \textdivide = {150,150}
7922 }
7923

```

7924 $\langle /ptm \rangle$

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).

```

7925  $\langle *!(blg|ugm) \rangle$ 
7926 \SetProtrusion
7927  $\langle m-t \rangle$  [ name = OT1-sc,
7928  $\langle bch \rangle$  [ name = bch-sc,
7929  $\langle cmr \rangle$  [ name = cmr-sc-OT1,
7930  $\langle ebg \rangle$  [ name = EBGaramond-sc-OT1-Prop,
7931  $\langle pmn \rangle$  [ name = pmnj-sc,
7932  $\langle ppl \rangle$  [ name = ppl-sc,
7933  $\langle ptm \rangle$  [ name = ptm-sc,
7934  $\langle m-t \rangle$  load = default ]
7935  $\langle bch \rangle$  load = bch-default ]
7936  $\langle cmr \rangle$  load = cmr-OT1 ]
7937  $\langle ebg \rangle$  load = EBGaramond-OT1-LF ]
7938  $\langle pmn \rangle$  load = pmnj-default ]
7939  $\langle ppl \rangle$  load = ppl-default ]
7940  $\langle ptm \rangle$  load = ptm-default ]
7941  $\langle m-t|bch|ebg|pmn \rangle$  { encoding = OT1,
7942  $\langle cmr|ppl|ptm \rangle$  { encoding = {OT1,OT4},
7943  $\langle bch \rangle$  family = bch,
7944  $\langle cmr \rangle$  family = cmr,
7945  $\langle ebg \rangle$  family = {EBGaramond-LF,EBGaramond-0sF},
7946  $\langle pmn \rangle$  family = pmnj,
7947  $\langle ppl \rangle$  family = {ppl,pplx,pplj},
7948  $\langle ptm \rangle$  family = {ptm,ptmx,ptmj},
7949 shape = sc }
7950 {
7951 a = {50,50},
7952  $\langle cmr|ebg|ppl|ptm \rangle$  \ae = {50, },
7953  $\langle bch|pmn \rangle$  c = {50, },
7954  $\langle bch|ebg|pmn \rangle$  d = { ,50},
7955  $\langle m-t|bch|cmr|ebg|pmn|ptm \rangle$  f = { ,50},
7956  $\langle bch|ebg|pmn \rangle$  g = {50, },
7957  $\langle m-t|cmr|ebg|pmn|ppl|ptm \rangle$  j = {50, },
7958  $\langle bch \rangle$  j = {100, },
7959  $\langle m-t|bch|cmr|ebg|pmn|ppl \rangle$  l = { ,50},
7960  $\langle ptm \rangle$  l = { ,80},
7961  $\langle m-t|bch|cmr|pmn|ppl \rangle$  013 = { ,50}, % f1
7962  $\langle ptm \rangle$  013 = { ,80}, % f1
7963  $\langle bch|ebg|pmn \rangle$  o = {50,50},
7964  $\langle ebg|pmn \rangle$  \oe = {50, },
7965  $\langle ppl \rangle$  p = { 0, 0},
7966  $\langle bch|ebg|pmn \rangle$  q = {50,70},
7967  $\langle ppl \rangle$  q = { 0, },
7968  $\langle m-t|cmr|ebg|pmn|ppl|ptm \rangle$  r = { , 0},
7969 t = {50,50},
7970  $\langle m-t|bch|cmr|ebg|pmn|ppl \rangle$  y = {50,50}
7971  $\langle ptm \rangle$  y = {80,80}
7972 }
7973
7974  $\langle *ebg \rangle$ 
7975 \SetProtrusion
7976 [ name = EBGaramond-sc-OT1-Tab,
7977 load = EBGaramond-OT1-T0sF ]
7978 { encoding = OT1,
7979 family = {EBGaramond-TLF,EBGaramond-T0sF},
7980 shape = sc }

```

```

7981 {
7982   a = {50,50},
7983   \ae = {50, },
7984   d = { ,50},
7985   f = { ,50},
7986   g = {50, },
7987   j = {50, },
7988   l = { ,50},
7989   o = {50,50},
7990   \oe = {50, },
7991   q = {50,70},
7992   r = { , 0},
7993   t = {50,50},
7994   y = {50,50}
7995 }
7996
7997 (/ebg)
7998 \SetProtrusion
7999 <m-t> [ name = Tl-sc,
8000 <bch> [ name = bch-sc-Tl,
8001 <cmr> [ name = cmr-sc-Tl,
8002 <ebg> [ name = EBGaramond-sc-Tl,
8003 <pmn> [ name = pmnj-sc-Tl,
8004 <ppl> [ name = ppl-sc-Tl,
8005 <ptm> [ name = ptm-sc-Tl,
8006 <m-t> load = Tl-default ]
8007 <bch> load = bch-Tl ]
8008 <cmr> load = cmr-Tl ]
8009 <ebg> load = EBGaramond-Tl ]
8010 <pmn> load = pmnj-Tl ]
8011 <ppl> load = ppl-Tl ]
8012 <ptm> load = ptm-Tl ]
8013 <!ebg> { encoding = {Tl,LYl},
8014 <ebg> { encoding = {LYl},
8015 <bch> family = bch,
8016 <cmr> family = cmr,
8017 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOsF},
8018 <pmn> family = pmnj,
8019 <ppl> family = {ppl,pplx,pplj},
8020 <ptm> family = {ptm,ptmx,ptmj},
8021 shape = sc }
8022 {
8023   a = {50,50},
8024 <cmr|ebg|ppl|ptm> \ae = {50, },
8025 <bch|pmn> c = {50, },
8026 <bch|ebg|pmn> d = { ,50},
8027 <m-t|bch|cmr|ebg|pmn|ptm> f = { ,50},
8028 <bch|ebg|pmn> g = {50, },
8029 <m-t|cmr|ebg|pmn|ppl|ptm> j = {50, },
8030 <bch> j = {100, },
8031 <m-t|bch|cmr|ebg|pmn|ppl> l = { ,50},
8032 <ptm> l = { ,80},
8033 <m-t|bch|cmr|pmn|ppl> 029 = { ,50}, % fl
8034 <ptm> 029 = { ,80}, % fl
8035 <bch|ebg|pmn> o = {50,50},
8036 <bch|ebg|pmn> \oe = {50, },
8037 <ppl> p = { 0, 0},
8038 <bch|ebg|pmn> q = {50,70},
8039 <ppl> q = { 0, },
8040 <m-t|cmr|ebg|pmn|ppl|ptm> r = { , 0},
8041 t = {50,50},
8042 <m-t|bch|cmr|ebg|pmn|ppl> y = {50,50}
8043 <ptm> y = {80,80}
8044 }
8045

```

```

8046 /(big|ugm)
8047 *m-t|cmr
8048 \SetProtrusion
8049 m-t [ name      = T2A-sc,
8050 cmr  [ name      = cmr-sc-T2A,
8051 m-t   load      = T2A-default ]
8052 cmr   load      = cmr-T2A   ]
8053 { encoding = T2A,
8054 cmr   family  = cmr,
8055   shape  = sc }
8056 {
8057   \cyra = {50,50},
8058   \cyrg = { ,50},
8059   \cyrt = {50,50},
8060   \cyry = { ,50}
8061 }
8062
8063 /m-t|cmr
8064 *m-t
8065 \SetProtrusion
8066 [ name      = QX-sc,
8067   load      = QX-default ]
8068 { encoding = QX,
8069   shape    = sc }
8070 {
8071   a = {50,50},
8072   f = { ,50},
8073   j = {50, },
8074   l = { ,50},
8075   013 = { ,50}, % fl
8076   r = { , 0},
8077   t = {50,50},
8078   y = {50,50}
8079 }
8080
8081 /m-t
8082 *cmr|bch
8083 \SetProtrusion
8084 bch [ name      = bch-sc-T5,
8085 bch  load      = bch-T5 ]
8086 cmr [ name      = cmr-sc-T5,
8087 cmr  load      = cmr-T5 ]
8088 { encoding = T5,
8089 bch   family  = bch,
8090 cmr   family  = cmr,
8091   shape  = sc }
8092 {
8093   a = {50,50},
8094 bch   c = {50, },
8095 bch   d = { ,50},
8096   f = { ,50},
8097 bch   g = {50, },
8098 bch   j = {100, },
8099 cmr   j = {50, },
8100   l = { ,50},
8101 bch   o = {50,50},
8102 bch   q = { 0, },
8103 cmr   r = { , 0},
8104   t = {50,50},
8105   y = {50,50}
8106 }
8107
8108 /cmr|bch
8109 *ebg
8110 \SetProtrusion

```

```

8111 [ name      = EBGaramond-sc-T1-Prop,
8112       load      = EBGaramond-T1-LF ]
8113 { encoding = T1,
8114   family   = {EBGaramond-LF,EBGaramond-OfF},
8115   shape    = sc }
8116 {
8117   a = {50,50},
8118   \ae = {50, },
8119   d = { ,50},
8120   f = { ,50},
8121   g = {50, },
8122   j = {50, },
8123   l = { ,50},
8124   o = {50,50},
8125   \oe = {50, },
8126   q = {50,70},
8127   r = { , 0},
8128   t = {50,50},
8129   y = {50,50}
8130 }
8131
8132 \SetProtrusion
8133 [ name      = EBGaramond-sc-T1-Tab,
8134       load      = EBGaramond-T1-TOfF ]
8135 { encoding = T1,
8136   family   = {EBGaramond-TLF,EBGaramond-TOfF},
8137   shape    = sc }
8138 {
8139   a = {50,50},
8140   \ae = {50, },
8141   d = { ,50},
8142   f = { ,50},
8143   g = {50, },
8144   j = {50, },
8145   l = { ,50},
8146   o = {50,50},
8147   \oe = {50, },
8148   q = {50,70},
8149   r = { , 0},
8150   t = {50,50},
8151   y = {50,50}
8152 }
8153
8154 </ebg>
8155 < *pmn >
8156 \SetProtrusion
8157 [ name      = pmnx-sc,
8158       load      = pmnj-sc ]
8159 { encoding = OT1,
8160   family   = pmnx,
8161   shape    = sc }
8162 {
8163   l = {230,180}
8164 }
8165
8166 \SetProtrusion
8167 [ name      = pmnx-sc-T1,
8168       load      = pmnj-sc-T1 ]
8169 { encoding = {T1,Ly1},
8170   family   = pmnx,
8171   shape    = sc }
8172 {
8173   l = {230,180}
8174 }
8175

```


2.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's font installation guide suggests `si`.

```

8176 \SetProtrusion
8177   [ name      = pmnj-scit,
8178     load      = pmnj-it   ]
8179   { encoding = OT1,
8180     family   = pmnj,
8181     shape    = {scit,si} }
8182   {
8183     a = {50, },
8184     \ae = { , -50},
8185     b = {20, -50},
8186     c = {50, -50},
8187     d = {20, 0},
8188     e = {20, -50},
8189     f = {10, 0},
8190     012 = {10, -50}, % fi
8191     013 = {10, -50}, % fl
8192     014 = {10, -50}, % ffi
8193     015 = {10, -50}, % ffl
8194     g = {50, -50},
8195     i = {20, -50},
8196     j = {20, 0},
8197     k = {20, },
8198     l = {20, 50},
8199     m = { , -30},
8200     n = { , -30},
8201     o = {50, },
8202     \oe = {50, -50},
8203     p = {20, -50},
8204     q = {50, },
8205     r = {20, 0},
8206     s = {20, -30},
8207     t = {70, },
8208     u = {50, -50},
8209     v = {100, },
8210     w = {100, },
8211     y = {50, },
8212     z = { , -50}
8213   }
8214
8215 \SetProtrusion
8216   [ name      = pmnj-scit-T1,
8217     load      = pmnj-it-T1 ]
8218   { encoding = {T1,LY1},
8219     family   = pmnj,
8220     shape    = {scit,si} }
8221   {
8222     a = {50, },
8223     \ae = { , -50},
8224     b = {20, -50},
8225     c = {50, -50},
8226     d = {20, 0},
8227     e = {20, -50},
8228     f = {10, 0},
8229     028 = {10, -50}, % fi
8230     029 = {10, -50}, % fl
8231     030 = {10, -50}, % ffi
8232     031 = {10, -50}, % ffl
8233     g = {50, -50},
8234     i = {20, -50},
8235     188 = {20, 0}, % ij
8236     j = {20, 0},

```

```

8237     k = {20, },
8238     l = {20,50},
8239     m = { , -30},
8240     n = { , -30},
8241     o = {50, },
8242     \oe = {50,-50},
8243     p = {20,-50},
8244     q = {50, },
8245     r = {20, 0},
8246     s = {20,-30},
8247     t = {70, },
8248     u = {50,-50},
8249     v = {100, },
8250     w = {100, },
8251     y = {50, },
8252     z = { , -50}
8253 }
8254
8255 \SetProtrusion
8256 [ name = pmnx-scit,
8257   load = pmnj-scit ]
8258 { encoding = OT1,
8259   family = pmnx,
8260   shape = {scit,si} }
8261 {
8262   l = {100,150}
8263 }
8264
8265 \SetProtrusion
8266 [ name = pmnx-scit-T1,
8267   load = pmnj-scit-T1 ]
8268 { encoding = {T1,LY1},
8269   family = pmnx,
8270   shape = {scit,si} }
8271 {
8272   l = {100,150}
8273 }
8274
8275 </pmn>
8276 <*ebg>

```

For small caps italics, we copy the definitions from the small caps settings, except that we first load the italics settings.

```

8277 \SetProtrusion
8278 [ name = EBGaramond-scit-OT1-Prop,
8279   load = EBGaramond-it-OT1-LF ]
8280 { encoding = OT1,
8281   family = {EBGaramond-LF,EBGaramond-0sF},
8282   shape = scit }
8283 {
8284   a = {50,50},
8285   \ae = {50, },
8286   d = { ,50},
8287   f = { ,50},
8288   g = {50, },
8289   j = {50, },
8290   l = { ,50},
8291   o = {50,50},
8292   \oe = {50, },
8293   q = {50,70},
8294   r = { , 0},
8295   t = {50,50},
8296   y = {50,50}
8297 }
8298

```

```
8299 \SetProtrusion
8300 [ name = EBGaramond-scit-OT1-Tab,
8301   load = EBGaramond-it-OT1-T0sF ]
8302 { encoding = OT1,
8303   family = {EBGaramond-TLF,EBGaramond-T0sF},
8304   shape = scit }
8305 {
8306   a = {50,50},
8307   \ae = {50, },
8308   d = { ,50},
8309   f = { ,50},
8310   g = {50, },
8311   j = {50, },
8312   l = { ,50},
8313   o = {50,50},
8314   \oe = {50, },
8315   q = {50,70},
8316   r = { , 0},
8317   t = {50,50},
8318   y = {50,50}
8319 }
8320
8321 \SetProtrusion
8322 [ name = EBGaramond-scit-T1-Prop,
8323   load = EBGaramond-it-T1-LF ]
8324 { encoding = T1,
8325   family = {EBGaramond-LF,EBGaramond-0sF},
8326   shape = scit }
8327 {
8328   a = {50,50},
8329   \ae = {50, },
8330   d = { ,50},
8331   f = { ,50},
8332   g = {50, },
8333   j = {50, },
8334   l = { ,50},
8335   o = {50,50},
8336   \oe = {50, },
8337   q = {50,70},
8338   r = { , 0},
8339   t = {50,50},
8340   y = {50,50}
8341 }
8342
8343 \SetProtrusion
8344 [ name = EBGaramond-scit-T1-Tab,
8345   load = EBGaramond-it-T1-T0sF ]
8346 { encoding = T1,
8347   family = {EBGaramond-TLF,EBGaramond-T0sF},
8348   shape = scit }
8349 {
8350   a = {50,50},
8351   \ae = {50, },
8352   d = { ,50},
8353   f = { ,50},
8354   g = {50, },
8355   j = {50, },
8356   l = { ,50},
8357   o = {50,50},
8358   \oe = {50, },
8359   q = {50,70},
8360   r = { , 0},
8361   t = {50,50},
8362   y = {50,50}
8363 }
```

8364
8365 */ebg*

2.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino.
Anybody?

```

8366 \SetProtrusion
8367 m-t [ name = textcomp ]
8368 bch [ name = bch-textcomp ]
8369 blg [ name = blg-textcomp ]
8370 cmr [ name = cmr-textcomp ]
8371 ebg [ name = EBGaramond-textcomp ]
8372 pnm [ name = pmn-textcomp ]
8373 ppl [ name = ppl-textcomp ]
8374 ptm [ name = ptm-textcomp ]
8375 ugm [ name = ugm-textcomp ]
8376 m-t { encoding = TS1 }
8377 !m-t { encoding = TS1,
8378 bch family = bch }
8379 blg family = blg }
8380 cmr family = cmr }
8381 ebg family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF} }
8382 pnm family = {pmnx,pmnj} }
8383 ppl family = {ppl,pplx,pplj} }
8384 ptm family = {ptm,ptmx,ptmj} }
8385 ugm family = ugm }
8386 {
8387 blg \textquotestraightbase = {400,500},
8388 cmr \textquotestraightbase = {300,300},
8389 ebg|pnm \textquotestraightbase = {400,400},
8390 blg \textquotestraightdblbase = {300,400},
8391 cmr|pnm \textquotestraightdblbase = {300,300},
8392 ebg \textquotestraightdblbase = {400,400},
8393 bch|cmr|ebg|pnm|ugm \texttwelveudash = {200,200},
8394 bch|cmr|ebg|pnm \textthreequartersemdash = {150,150},
8395 ugm \textthreequartersemdash = {200,200},
8396 blg \textquotesingle = {500,600},
8397 cmr|pnm \textquotesingle = {300,400},
8398 ebg \textquotesingle = {400,500},
8399 ptm \textquotesingle = {500,500},
8400 ugm \textquotesingle = {300,500},
8401 bch|cmr|pnm \textasteriskcentered = {200,300},
8402 blg \textasteriskcentered = {150,200},
8403 ebg \textasteriskcentered = {300,300},
8404 ugm \textasteriskcentered = {100,200},
8405 pnm \textfractionsolidus = {-200,-200},
8406 cmr \textoneoldstyle = {100,100},
8407 pnm \textoneoldstyle = { , 50},
8408 cmr \textthreeoldstyle = { , 50},
8409 ebg|pnm \textthreeoldstyle = { 50, },
8410 cmr \textfouroldstyle = { 50, 50},
8411 ebg|pnm \textfouroldstyle = { 50, },
8412 cmr|ebg|pnm \textsevenoldstyle = { 50, 80},
8413 cmr \textlangle = {400, },
8414 cmr \textrangle = { ,400},
8415 m-t|bch|pnm|ptm \textminus = {200,200},
8416 cmr|ebg|ppl \textminus = {300,300},
8417 blg|ugm \textminus = {250,300},
8418 bch|ebg|pnm \textlbrackdbl = {100, },
8419 blg \textlbrackdbl = {200, },
8420 bch|ebg|pnm \textrbrackdbl = { ,100},
8421 blg \textrbrackdbl = { ,200},
8422 pnm \textasciigrave = {200,500},

```

```

8423 <bch|blg|cmr|ebg|pmn> \texttildelow = {200,250},
8424 <pmn> \textasciibreve = {300,400},
8425 <pmn> \textasciicaron = {300,400},
8426 <pmn> \textacutedbl = {200,300},
8427 <pmn> \textgravedbl = {150,300},
8428 <bch|pmn|ugm> \textdagger = { 80, 80},
8429 <blg> \textdagger = {200,200},
8430 <cmr|ebg> \textdagger = {100,100},
8431 <ptm> \textdagger = {150,150},
8432 <blg> \textdaggerdbl = {150,150},
8433 <cmr|ebg|pmn> \textdaggerdbl = { 80, 80},
8434 <ptm> \textdaggerdbl = {100,100},
8435 <bch> \textbardbl = {100,100},
8436 <blg|ugm> \textbardbl = {150,150},
8437 <bch> \textbullet = {200,200},
8438 <blg> \textbullet = {400,500},
8439 <cmr|ebg|pmn> \textbullet = { ,100},
8440 <ptm> \textbullet = {150,150},
8441 <ugm> \textbullet = { 50,100},
8442 <bch|cmr|pmn> \textcelsius = { 50, },
8443 <ebg> \textcelsius = { 80, },
8444 <bch> \textflorin = { 50, 50},
8445 <blg> \textflorin = {100,100},
8446 <ebg|ugm> \textflorin = { ,100},
8447 <pmn> \textflorin = { 50,100},
8448 <ptm> \textflorin = { 50, 70},
8449 <cmr> \textcolonmonetary = { , 50},
8450 <ebg|pmn> \textcolonmonetary = { 50, },
8451 <pmn> \textinterrobang = { ,100},
8452 <pmn> \textinterrobangdown = {100, },
8453 <m-t|ebg|ptm> \texttrademark = {100,100},
8454 <bch> \texttrademark = {150,150},
8455 <blg|cmr|ppl> \texttrademark = {200,200},
8456 <pmn> \texttrademark = { 50, 50},
8457 <ugm> \texttrademark = {100,150},
8458 <bch|ugm> \textcent = { 50, },
8459 <ptm> \textcent = {100,100},
8460 <bch> \textsterling = { 50, },
8461 <ugm> \textsterling = { , 50},
8462 <bch> \textbrokenbar = {200,200},
8463 <blg> \textbrokenbar = {250,250},
8464 <ugm> \textbrokenbar = {200,300},
8465 <pmn> \textasciidieresis = {300,400},
8466 <m-t|bch|cmr|ebg|ptm|ugm> \textcopyright = {100,100},
8467 <pmn> \textcopyright = {100,150},
8468 <ppl> \textcopyright = {200,200},
8469 <bch|cmr|ugm> \textordfeminine = {100,200},
8470 <ebg|pmn> \textordfeminine = {200,200},
8471 <bch|cmr|ebg|pmn|ugm> \textlnot = {200, },
8472 <blg> \textlnot = {200,100},
8473 <m-t|bch|cmr|ebg|ptm|ugm> \textregistered = {100,100},
8474 <pmn> \textregistered = { 50,150},
8475 <ppl> \textregistered = {200,200},
8476 <pmn> \textasciimacron = {150,200},
8477 <m-t|ppl|ptm> \textdegree = {300,300},
8478 <bch> \textdegree = {150,200},
8479 <blg|ugm> \textdegree = {200,200},
8480 <cmr|ebg> \textdegree = {400,400},
8481 <pmn> \textdegree = {150,400},
8482 <bch|cmr|ebg|pmn|ugm> \textpm = {150,200},
8483 <blg> \textpm = {100,100},
8484 <ptm> \textpm = { 50, 80},
8485 <bch|blg|ugm> \texttwosuperior = {100,200},
8486 <cmr> \texttwosuperior = { 50,100},
8487 <ebg|pmn> \texttwosuperior = {200,200},

```

```

8488 <ptm> \texttwosuperior = { 50, 50},
8489 <bch|blg|ugm> \textthreesuperior = {100,200},
8490 <cmr> \textthreesuperior = { 50,100},
8491 <ebg|pmn> \textthreesuperior = {200,200},
8492 <ptm> \textthreesuperior = { 50, 50},
8493 <pmn> \textasciicute = {300,400},
8494 <bch|ugm> \textmu = { ,100},
8495 <bch|ebg|pmn> \textparagraph = { ,100},
8496 <bch|cmr|ebg|pmn> \textperiodcentered = {300,400},
8497 <blg> \textperiodcentered = {400,500},
8498 <ptm> \textperiodcentered = {300,300},
8499 <ugm> \textperiodcentered = {200,500},
8500 <bch|blg|ugm> \textonesuperior = {200,300},
8501 <cmr|ebg|pmn> \textonesuperior = {200,200},
8502 <ptm> \textonesuperior = {100,100},
8503 <bch|ebg|pmn|ugm> \textordmasculine = {200,200},
8504 <blg|cmr> \textordmasculine = {100,200},
8505 <bch|cmr|pmn> \texteuro = {100, },
8506 <ebg> \texteuro = { 50,100},
8507 <bch> \texttimes = {200,200},
8508 <blg|ptm> \texttimes = {100,100},
8509 <cmr> \texttimes = {150,250},
8510 <ebg> \texttimes = {100,150},
8511 <pmn> \texttimes = { 70,100},
8512 <ugm> \texttimes = {200,300},
8513 <bch|ebg|pmn> \textdiv = {150,200}
8514 <blg> \textdiv = {100,100}
8515 <cmr> \textdiv = {150,250}
8516 <ptm> \textdiv = { 50,100},
8517 <ugm> \textdiv = {200,300},
8518 <ptm> \textperthousand = { ,50}
8519 <ugm> \textsection = { ,100},
8520 <ugm> \textonehalf = { 50,100},
8521 <ugm> \textonequarter = { 50,100},
8522 <ugm> \textthreequarters = { 50,100},
8523 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

8524 }
8525
8526 <*cmr|ebg|pmn|ugm>
8527 \SetProtrusion
8528 <cmr> [ name = cmr-textcomp-it ]
8529 <ebg> [ name = EBGaramond-textcomp-it ]
8530 <pmn> [ name = pmn-textcomp-it ]
8531 <ugm> [ name = ugm-textcomp-it ]
8532 { encoding = TS1,
8533 <cmr> family = cmr,
8534 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
8535 <pmn> family = {pmnx,pmnj},
8536 <ugm> family = ugm,
8537 <cmr|pmn> shape = {it,s1} }
8538 <ebg|ugm> shape = it }
8539 {
8540 <cmr> \textquotestraightbase = {300,600},
8541 <ebg|pmn> \textquotestraightbase = {400,400},
8542 <cmr> \textquotestraightdblbase = {300,600},
8543 <ebg> \textquotestraightdblbase = {300,400},
8544 <pmn> \textquotestraightdblbase = {300,300},
8545 \texttwelveudash = {200,200},
8546 <cmr|ebg|pmn> \textthreequartersemdash = {150,150},
8547 <ugm> \textthreequartersemdash = {200,200},
8548 <cmr> \textquotesingle = {600,300},
8549 <ebg> \textquotesingle = {800,100},
8550 <pmn> \textquotesingle = {300,200},

```

```

8551 <ugm> \textquotesingle = {500,500},
8552 <cmr> \textasteriskcentered = {300,200},
8553 <ebg> \textasteriskcentered = {500,100},
8554 <pmn> \textasteriskcentered = {200,300},
8555 <ugm> \textasteriskcentered = {300,150},
8556 <pmn> \textfractionsolidus = {-200,-200},
8557 <cmr> \textoneoldstyle = {100, 50},
8558 <ebg> \textoneoldstyle = {100, },
8559 <pmn> \textoneoldstyle = { 50, },
8560 <ebg> \texttwooldstyle = { 50, },
8561 <pmn> \texttwooldstyle = {-50, },
8562 <cmr> \textthreeoldstyle = {100, 50},
8563 <pmn> \textthreeoldstyle = {-100, },
8564 <cmr> \textfouroldstyle = { 50, 50},
8565 <ebg> \textfouroldstyle = { 50,100},
8566 <cmr> \textsevenoldstyle = { 50, 80},
8567 <ebg> \textsevenoldstyle = { 50, },
8568 <pmn> \textsevenoldstyle = { 20, },
8569 <cmr> \textlangle = {400, },
8570 <cmr> \textrightangle = { ,400},
8571 <cmr|ebg> \textminus = = {300,300},
8572 <pmn> \textminus = {200,200},
8573 <ugm> \textminus = {250,300},
8574 <ebg|pmn> \textlbrackdbl = {100, },
8575 <ebg|pmn> \textrbrackdbl = { ,100},
8576 <pmn> \textasciigrave = {300,300},
8577 <cmr|ebg|pmn> \texttildebelow = {200,250},
8578 <pmn> \textasciibreve = {300,300},
8579 <pmn> \textasciicaron = {300,300},
8580 <pmn> \textacutedbl = {200,300},
8581 <pmn> \textgravedbl = {150,300},
8582 <cmr> \textdagger = {100,100},
8583 <ebg> \textdagger = {200,100},
8584 <pmn> \textdagger = { 80, 50},
8585 <ugm> \textdagger = { 80, 80},
8586 <cmr|ebg> \textdaggerdbl = = { 80, 80},
8587 <pmn> \textdaggerdbl = { 80, 50},
8588 <ugm> \textbardbl = {150,150},
8589 <cmr> \textbullet = {200,100},
8590 <ebg> \textbullet = {300, },
8591 <pmn> \textbullet = { 30, 70},
8592 <ugm> \textbullet = { 50,100},
8593 <cmr> \textcelsius = {100, },
8594 <ebg> \textcelsius = {200, },
8595 <pmn> \textcelsius = { 50,-50},
8596 <ebg> \textflorin = {100, },
8597 <pmn> \textflorin = { 50,100},
8598 <ugm> \textflorin = { ,100},
8599 <cmr> \textcolonmonetary = {150, },
8600 <ebg> \textcolonmonetary = {100, },
8601 <pmn> \textcolonmonetary = { 50,-50},
8602 <cmr|ebg> \texttrademark = = {200, },
8603 <pmn> \texttrademark = { 50,100},
8604 <ugm> \texttrademark = {150, 50},
8605 <ugm> \textcent = { 50, },
8606 <ugm> \textsterling = { , 50},
8607 <ugm> \textbrokenbar = {200,300},
8608 <pmn> \textasciidieresis = {300,200},
8609 <cmr> \textcopyright = {100, },
8610 <ebg> \textcopyright = {200,100},
8611 <pmn> \textcopyright = {100,150},
8612 <ugm> \textcopyright = {300, },
8613 <cmr> \textordfeminine = {100,100},
8614 <pmn> \textordfeminine = {200,200},
8615 <ugm> \textordfeminine = {100,200},

```

```

8616 <cmr|ebg> \textlnot = {300, },
8617 <pmn|ugm> \textlnot = {200, },
8618 <cmr> \textregistered = {100, },
8619 <ebg> \textregistered = {200,100},
8620 <pmn> \textregistered = { 50,150},
8621 <ugm> \textregistered = {300, },
8622 <pmn> \textasciimacron = {150,200},
8623 <cmr|ebg> \textdegree = {500,100},
8624 <pmn> \textdegree = {150,150},
8625 <ugm> \textdegree = {300,200},
8626 <cmr> \textpm = {150,100},
8627 <ebg> \textpm = {200,150},
8628 <pmn|ugm> \textpm = {150,200},
8629 <cmr> \textonesuperior = {400, },
8630 <ebg> \textonesuperior = {300,100},
8631 <pmn> \textonesuperior = {200,100},
8632 <ugm> \textonesuperior = {300,300},
8633 <cmr> \texttwosuperior = {400, },
8634 <ebg> \texttwosuperior = {300, },
8635 <pmn> \texttwosuperior = {200,100},
8636 <ugm> \texttwosuperior = {300,200},
8637 <cmr> \textthreesuperior = {400, },
8638 <ebg> \textthreesuperior = {300, },
8639 <pmn> \textthreesuperior = {200,100},
8640 <ugm> \textthreesuperior = {300,200},
8641 <ugm> \textmu = { ,100},
8642 <pmn> \textasciiaacute = {300,200},
8643 <cmr> \textparagraph = {200, },
8644 <pmn> \textparagraph = { ,100},
8645 <cmr> \textperiodcentered = {500,500},
8646 <ebg|pmn|ugm> \textperiodcentered = {300,400},
8647 <cmr> \textordmasculine = {100,100},
8648 <pmn> \textordmasculine = {200,200},
8649 <ugm> \textordmasculine = {300,200},
8650 <cmr> \texteuro = {200, },
8651 <ebg> \texteuro = {100, },
8652 <pmn> \texteuro = {100,-50},
8653 <cmr> \texttimes = {200,200},
8654 <ebg> \texttimes = {200,100},
8655 <pmn> \texttimes = { 70,100},
8656 <ugm> \texttimes = {200,300},
8657 <cmr|ebg> \textdiv = {200,200}
8658 <pmn> \textdiv = {150,200}
8659 <ugm> \textdiv = {200,300},
8660 <ugm> \textsection = { ,200},
8661 <ugm> \textonehalf = { 50,100},
8662 <ugm> \textonequarter = { 50,100},
8663 <ugm> \textthreequarters = { 50,100},
8664 <ugm> \textsurd = { ,100}
8665 }
8666
8667 </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}
```

```
8668 (*cmr)
8669 \SetProtrusion
8670 [ name      = cmr-math-letters ]
8671 { encoding = OML,
8672   family   = cmm,
8673   series    = {m,b},
8674   shape     = it   }
8675 {
8676   A = {100, 50}, % \mathnormal
8677   B = { 50,   },
8678   C = { 50,   },
8679   D = { 50, 50},
8680   E = { 50,   },
8681   F = {100, 50},
8682   G = { 50, 50},
8683   H = { 50, 50},
8684   I = { 50, 50},
8685   J = {150, 50},
8686   K = { 50,100},
8687   L = { 50, 50},
8688   M = { 50,   },
8689   N = { 50,   },
8690   O = { 50,   },
8691   P = { 50,   },
8692   Q = { 50, 50},
8693   R = { 50,   },
8694   S = { 50,   },
8695   T = { 50,100},
8696   U = { 50, 50},
8697   V = {100,100},
8698   W = { 50,100},
8699   X = { 50,100},
8700   Y = {100,100},
8701   f = {100,100},
8702   h = {   ,100},
8703   i = {   , 50},
8704   j = {   , 50},
8705   k = {   , 50},
8706   r = {   , 50},
8707   v = {   , 50},
8708   w = {   , 50},
8709   x = {   , 50},
8710   "0B = { 50,100}, % \alpha
8711   "0C = { 50, 50}, % \beta
8712   "0D = {200,150}, % \gamma
8713   "0E = { 50, 50}, % \delta
8714   "0F = { 50, 50}, % \epsilon
8715   "10 = { 50,150}, % \zeta
8716   "12 = { 50,   }, % \theta
8717   "13 = {   ,100}, % \iota
8718   "14 = {   ,100}, % \kappa
8719   "15 = {100, 50}, % \lambda
8720   "16 = {   , 50}, % \mu
8721   "17 = {   , 50}, % \nu
8722   "18 = {   , 50}, % \xi
8723   "19 = { 50,100}, % \pi
8724   "1A = { 50, 50}, % \rho
8725   "1B = {   ,150}, % \sigma
```

```

8726 "1C = { 50,150}, % \tau
8727 "1D = { 50, 50}, % \upsilon
8728 "1F = { 50,100}, % \chi
8729 "20 = { 50, 50}, % \psi
8730 "21 = { , 50}, % \omega
8731 "22 = { , 50}, % \varepsilon
8732 "23 = { , 50}, % \vartheta
8733 "24 = { , 50}, % \varphi
8734 "25 = {100, }, % \varrho
8735 "26 = {100,100}, % \varsigma
8736 "27 = { 50, 50}, % \varphi
8737 "28 = {100,100}, % \leftharpoonup
8738 "29 = {100,100}, % \leftharpoondown
8739 "2A = {100,100}, % \rightharpoonup
8740 "2B = {100,100}, % \rightharpoondown
8741 "2C = {300,200}, % \lhook
8742 "2D = {200,300}, % \rhook
8743 "2E = { ,100}, % \triangleright
8744 "2F = {100, }, % \triangleleft
8745 "3A = { ,500}, % \cdot
8746 "3B = { ,500}, % ,
8747 "3C = {200,100}, % <
8748 "3D = {300,400}, % /
8749 "3E = {100,200}, % >
8750 "3F = {200,200}, % \star
8751 "5B = { ,100}, % \flat
8752 "5E = {200,200}, % \smile
8753 "5F = {200,200}, % \frown
8754 "7C = {100, }, % \jmath
8755 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

8756 }
8757

```

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}

```

```

8758 \SetProtrusion
8759 [ name = cmr-math-symbols ]
8760 { encoding = OMS,
8761 family = cmsy,
8762 series = {m,b},
8763 shape = n }
8764 {
8765 A = {150, 50}, % \mathcal
8766 C = { ,100},
8767 D = { , 50},
8768 F = { 50,150},
8769 I = { ,100},
8770 J = {100,150},
8771 K = { ,100},
8772 L = {100, },
8773 M = { 50, 50},
8774 N = { 50,100},
8775 P = { , 50},
8776 Q = { 50, },
8777 R = { , 50},
8778 T = { 50,150},
8779 V = { 50, 50},
8780 W = { , 50},
8781 X = {100,100},
8782 Y = {100, },
8783 Z = {100,150},

```

```

8784 "00 = {300,300}, % -
8785 "01 = { ,700}, % \cdot, \cdotp
8786 "02 = {150,250}, % \times
8787 "03 = {150,250}, % *, \ast
8788 "04 = {200,300}, % \div
8789 "05 = {150,250}, % \diamond
8790 "06 = {200,200}, % \mp
8791 "07 = {200,200}, % \mp
8792 "08 = {100,100}, % \oplus
8793 "09 = {100,100}, % \ominus
8794 "0A = {100,100}, % \otimes
8795 "0B = {100,100}, % \oslash
8796 "0C = {100,100}, % \odot
8797 "0D = {100,100}, % \bigcirc
8798 "0E = {100,100}, % \circ
8799 "0F = {100,100}, % \bullet
8800 "10 = {100,100}, % \asymp
8801 "11 = {100,100}, % \equiv
8802 "12 = {200,100}, % \subseteq
8803 "13 = {100,200}, % \supseteq
8804 "14 = {200,100}, % \leq
8805 "15 = {100,200}, % \geq
8806 "16 = {200,100}, % \preceq
8807 "17 = {100,200}, % \succeq
8808 "18 = {200,200}, % \sim
8809 "19 = {150,150}, % \approx
8810 "1A = {200,100}, % \subset
8811 "1B = {100,200}, % \supset
8812 "1C = {200,100}, % \ll
8813 "1D = {100,200}, % \gg
8814 "1E = {300,100}, % \prec
8815 "1F = {100,300}, % \succ
8816 "20 = {100,200}, % \leftarrow
8817 "21 = {200,100}, % \rightarrow
8818 "22 = {100,100}, % \uparrow
8819 "23 = {100,100}, % \downarrow
8820 "24 = {100,100}, % \leftrightarrows
8821 "25 = {100,100}, % \nearrow
8822 "26 = {100,100}, % \searrow
8823 "27 = {100,100}, % \simeq
8824 "28 = {100,100}, % \Leftarrow
8825 "29 = {100,100}, % \Rightarrow
8826 "2A = {100,100}, % \Uparrow
8827 "2B = {100,100}, % \Downarrow
8828 "2C = {100,100}, % \Leftrightarrow
8829 "2D = {100,100}, % \nrightarrow
8830 "2E = {100,100}, % \swarrow
8831 "2F = { ,100}, % \propto
8832 "30 = { ,400}, % \prime
8833 "31 = {100,100}, % \infty
8834 "32 = {150,100}, % \in
8835 "33 = {100,150}, % \ni
8836 "34 = {100,100}, % \triangle, \bigtriangleup
8837 "35 = {100,100}, % \bigtriangledown
8838 "38 = { ,100}, % \forall
8839 "39 = {100, }, % \exists
8840 "3A = {200, }, % \neg
8841 "3E = {200,200}, % \top
8842 "3F = {200,200}, % \bot, \perp
8843 "5E = {100,200}, % \wedge
8844 "5F = {100,200}, % \vee
8845 "60 = { ,300}, % \vdash
8846 "61 = {300, }, % \dashv
8847 "62 = {100,100}, % \lfloor
8848 "63 = {100,100}, % \rfloor

```

```

8849 "64 = {100,100}, % \lceil
8850 "65 = {100,100}, % \rceil
8851 "66 = {150, }, % \lbrace
8852 "67 = { ,150}, % \rbrace
8853 "68 = {400, }, % \langle
8854 "69 = { ,400}, % \rangle
8855 "6C = {100,100}, % \updownarrow
8856 "6D = {100,100}, % \Updownarrow
8857 "6E = {100,300}, % \, \backslash, \setminus
8858 "72 = {100,100}, % \nabla
8859 "79 = {200,200}, % \dagger
8860 "7A = {100,100}, % \ddagger
8861 "7B = {100, }, % \mathparagraph
8862 "7C = {100,100}, % \clubsuit
8863 "7D = {100,100}, % \diamondsuit
8864 "7E = {100,100}, % \heartsuit
8865 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

8866 }
8867

```

We don't bother about 'largsymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largsymbols}{OMX}{cmex}{m}{n}
```

```

8868 </cmr>
8869 </cfg-t>

```

2.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
8870 <*cfg-u>
```

Symbol font 'a'.

```

8871 <*msa>
8872 \SetProtrusion
8873 [ name = AMS-a ]
8874 { encoding = U,
8875   family = msa }
8876 {
8877 "05 = {150,250}, % \centerdot
8878 "06 = {100,100}, % \lozenge
8879 "07 = { 50, 50}, % \blacklozenge
8880 "08 = { 50, 50}, % \circlearrowright
8881 "09 = { 50, 50}, % \circlearrowleft
8882 "0A = {100,100}, % \rightleftharpoons
8883 "0B = {100,100}, % \leftrightharpoons
8884 "0D = {-50,200}, % \Vdash
8885 "0E = {-50,200}, % \Vvdash
8886 "0F = {-70,150}, % \vDash
8887 "10 = {100,150}, % \twoheadrightarrow
8888 "11 = {100,150}, % \twoheadleftarrow
8889 "12 = { 50,100}, % \leftleftarrows
8890 "13 = { 50, 80}, % \rightrightarrows
8891 "14 = {120,120}, % \upuparrows
8892 "15 = {120,120}, % \downdownarrows
8893 "16 = {200,200}, % \upharpoonright
8894 "17 = {200,200}, % \downharpoonright
8895 "18 = {200,200}, % \upharpoonleft
8896 "19 = {200,200}, % \downharpoonleft
8897 "1A = { 80,100}, % \rightarrowtail
8898 "1B = { 80,100}, % \leftarrowtail

```

```

8899 "1C = { 50, 50}, % \leftrightharrows
8900 "1D = { 50, 50}, % \rightleftarrows
8901 "1E = {250, }, % \Lsh
8902 "1F = { ,250}, % \Rsh
8903 "20 = {100,100}, % \rightsquigarrow
8904 "21 = {100,100}, % \leftrightsquigarrow
8905 "22 = {100, 50}, % \looparrowleft
8906 "23 = { 50,100}, % \looparrowright
8907 "24 = { 50, 80}, % \circeq
8908 "25 = { ,100}, % \succsim
8909 "26 = { ,100}, % \gtrsim
8910 "27 = { ,100}, % \gtrapprox
8911 "28 = {150, 50}, % \multimap
8912 "2B = {100,150}, % \doteqdot
8913 "2C = {100,150}, % \triangleq
8914 "2D = {100, 50}, % \precsim
8915 "2E = {100, 50}, % \lessim
8916 "2F = { 50, 50}, % \lessapprox
8917 "30 = {100, 50}, % \eqslantless
8918 "31 = { 50, 50}, % \eqslantgtr
8919 "32 = {100, 50}, % \curlyeqprec
8920 "33 = { 50,100}, % \curlyeqsucc
8921 "34 = {100, 50}, % \preccurlyeq
8922 "36 = { 50, }, % \leqslant
8923 "38 = { , 50}, % \backprime
8924 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
8925 "3C = { 50,100}, % \succcurlyeq
8926 "3E = { , 50}, % \geqslant
8927 "40 = { , 50}, % \sqsubset
8928 "41 = { 50, }, % \sqsupset
8929 "42 = { ,150}, % \vartriangleright, \rhd
8930 "43 = {150, }, % \vartriangleleft, \lhd
8931 "44 = { ,100}, % \trianglerighteq, \unrhd
8932 "45 = {100, }, % \trianglelefteq, \unlhd
8933 "46 = {100,100}, % \bigstar
8934 "48 = { 50, 50}, % \blacktriangledown
8935 "49 = { ,100}, % \blacktriangleright
8936 "4A = {100, }, % \blacktriangleleft
8937 "4B = { ,150}, % \dashrightarrow (the arrow)
8938 "4C = {150, }, % \dashleftarrow
8939 "4D = { 50, 50}, % \vartriangle
8940 "4E = { 50, 50}, % \blacktriangle
8941 "4F = { 50, 50}, % \triangledown
8942 "50 = { 50, 50}, % \eqcirc
8943 "56 = { ,150}, % \Rrightarrow
8944 "57 = {150, }, % \Lleftarrow
8945 "58 = {100,300}, % \checkmark
8946 "5C = { 50, 50}, % \angle
8947 "5D = { 50, 50}, % \measuredangle
8948 "5E = { 50, 50}, % \sphericalangle
8949 "5F = { , 50}, % \varpropto
8950 "60 = {100,100}, % \smallsmile
8951 "61 = {100,100}, % \smallfrown
8952 "62 = { 50, }, % \Subset
8953 "63 = { , 50}, % \Supset
8954 "66 = {150,150}, % \curlywedge
8955 "67 = {150,150}, % \curlyvee
8956 "68 = { 50,150}, % \leftthreetimes
8957 "69 = {100, 50}, % \rightthreetimes
8958 "6C = { 50, 50}, % \bumpeq
8959 "6D = { 50, 50}, % \Bumpeq
8960 "6E = {100, }, % \lll
8961 "6F = { ,100}, % \ggg
8962 "70 = { 50,100}, % \ulcorner
8963 "71 = {100, 50}, % \urcorner

```

```

8964 "75 = {150,200}, % \dotplus
8965 "76 = { 50,100}, % \backsim
8966 "78 = { 50,100}, % \llcorner
8967 "79 = {100, 50}, % \lrcorner
8968 "7C = {100,100}, % \intercal
8969 "7D = { 50, 50}, % \circledcirc
8970 "7E = { 50, 50}, % \circledast
8971 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

8972 }
8973
8974 </msa>

```

Symbol font 'b'.

```

8975 <*msb>
8976 \SetProtrusion
8977 [ name = AMS-b ]
8978 { encoding = U,
8979   family = msb }
8980 {
8981   A = { 50, 50}, % \mathbb
8982   C = { 50, 50},
8983   G = { , 50},
8984   L = { , 50},
8985   P = { , 50},
8986   R = { , 50},
8987   T = { , 50},
8988   V = { 50, 50},
8989   X = { 50, 50},
8990   Y = { 50, 50},
8991 "00 = { 50, 50}, % \lvertneqq
8992 "01 = { 50, 50}, % \gvertneqq
8993 "02 = { 50, 50}, % \lneq
8994 "03 = { 50, 50}, % \gneq
8995 "04 = {100, 50}, % \lless
8996 "05 = { 50,150}, % \ngtr
8997 "06 = {100, 50}, % \nprec
8998 "07 = { 50,150}, % \nsucc
8999 "08 = { 50, 50}, % \lneqq
9000 "09 = { 50, 50}, % \gneqq
9001 "0A = {100,100}, % \lneqslant
9002 "0B = {100,100}, % \gneqslant
9003 "0C = {100, 50}, % \lneq
9004 "0D = { 50,100}, % \gneq
9005 "0E = {100, 50}, % \npreceq
9006 "0F = { 50,100}, % \nsucceq
9007 "10 = { 50, }, % \precnsim
9008 "11 = { 50, 50}, % \succnsim
9009 "12 = { 50, 50}, % \lnsim
9010 "13 = { 50, 50}, % \gnsim
9011 "14 = { 50, 50}, % \lneqq
9012 "15 = { 50, 50}, % \gneqq
9013 "16 = { 50, 50}, % \precneqq
9014 "17 = { 50, 50}, % \succneqq
9015 "18 = { 50, 50}, % \precnapprox
9016 "19 = { 50, 50}, % \succnapprox
9017 "1A = { 50, 50}, % \lnaprox
9018 "1B = { 50, 50}, % \gnapprox
9019 "1C = {150,200}, % \nsim
9020 "1D = { 50, 50}, % \ncong
9021 "1E = {100,150}, % \diagup
9022 "1F = {100,150}, % \diagdown
9023 "20 = {100, 50}, % \varsubsetneq
9024 "21 = { 50,100}, % \varsupsetneq

```

```

9025 "22 = {100, 50}, % \subsetteqq
9026 "23 = { 50,100}, % \supsetteqq
9027 "24 = {100, 50}, % \subsetneqq
9028 "25 = { 50,100}, % \supsetneqq
9029 "26 = {100, 50}, % \varsubsetneqq
9030 "27 = { 50,100}, % \varsupsetneqq
9031 "28 = {100, 50}, % \subseteq
9032 "29 = { 50,100}, % \supseteq
9033 "2A = {100, 50}, % \subset
9034 "2B = { 50,100}, % \supset
9035 "2C = { 50,100}, % \parallel
9036 "2D = {100,150}, % \mid
9037 "2E = {150,150}, % \shortmid
9038 "2F = {100,100}, % \shortparallel
9039 "30 = { ,150}, % \ndash
9040 "31 = { ,150}, % \vdash
9041 "32 = { ,100}, % \vDash
9042 "33 = { ,100}, % \Vdash
9043 "34 = { ,100}, % \trianglerighteq
9044 "35 = {100, }, % \trianglelefteq
9045 "36 = {100, }, % \triangleleft
9046 "37 = { ,100}, % \triangleright
9047 "38 = {100,200}, % \leftarrow
9048 "39 = {100,200}, % \rightarrow
9049 "3A = {100,100}, % \Leftarrow
9050 "3B = { 50,100}, % \Rightarrow
9051 "3C = {100,100}, % \Leftrightarrow
9052 "3D = {100,200}, % \leftrightarrows
9053 "3E = { 50, 50}, % \divideontimes
9054 "3F = { 50, 50}, % \varnothing
9055 "60 = {200, }, % \Finv
9056 "61 = { , 50}, % \Game
9057 "68 = {100,100}, % \eqsim
9058 "69 = { 50, }, % \beth
9059 "6A = { 50, }, % \gimel
9060 "6B = {150, }, % \daleth
9061 "6C = {200, }, % \lessdot
9062 "6D = { ,200}, % \gtrdot
9063 "6E = {100,200}, % \ltimes
9064 "6F = {150,100}, % \rtimes
9065 "70 = { 50,100}, % \shortmid
9066 "71 = { 50, 50}, % \shortparallel
9067 "72 = {200,300}, % \smallsetminus
9068 "73 = {100,200}, % \thicksim
9069 "74 = { 50,100}, % \thickapprox
9070 "75 = { 50, 50}, % \approx
9071 "76 = { 50,100}, % \succapprox
9072 "77 = { 50, 50}, % \preccurlyeq
9073 "78 = {100,100}, % \curvearrowleft
9074 "79 = { 50,150}, % \curvearrowright
9075 "7A = { 50,200}, % \digamma
9076 "7B = {100, 50}, % \varkappa
9077 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

9078 }
9079
9080 (/msb)

```

2.8.8 Euler

Euler Roman font (package `euler`).

```

9081 (*eur)
9082 \SetProtrusion

```

```

9083 [ name      = euler ]
9084 { encoding = U,
9085   family   = eur  }
9086 {
9087   "01 = {100,100},
9088   "03 = {100,150},
9089   "06 = {   ,100},
9090   "07 = {100,150},
9091   "08 = {100,100},
9092   "0A = {100,100},
9093   "0B = {   , 50},
9094   "0C = {   ,100},
9095   "0D = {100,100},
9096   "0E = {   ,100},
9097   "0F = {100,100},
9098   "10 = {100,100},
9099   "13 = {   ,100},
9100   "14 = {   ,100},
9101   "15 = {   , 50},
9102   "16 = {   , 50},
9103   "17 = { 50,100},
9104   "18 = { 50,100},
9105   "1A = {   , 50},
9106   "1B = {   , 50},
9107   "1C = { 50,100},
9108   "1D = { 50,100},
9109   "1E = { 50,100},
9110   "1F = { 50,100},
9111   "20 = {   , 50},
9112   "21 = {   , 50},
9113   "22 = { 50,100},
9114   "24 = {   , 50},
9115   "27 = { 50,100},
9116   1  = {100,100},
9117   7  = { 50,100},
9118   "3A = {300,500},
9119   "3B = {200,400},
9120   "3C = {200,100},
9121   "3D = {200,200},
9122   "3E = {100,200},
9123   A  = {   ,100},
9124   D  = {   , 50},
9125   J  = { 50,   },
9126   K  = {   , 50},
9127   L  = {   , 50},
9128   Q  = {   , 50},
9129   T  = { 50,   },
9130   X  = { 50, 50},
9131   Y  = { 50,   },
9132   h  = {   , 50},
9133   k  = {   , 50}
9134 }
9135

```

Extended by the eulervm package.

```

9136 \SetProtrusion
9137 [ name      = euler-vm,
9138   load      = euler ]
9139 { encoding = U,
9140   family   = zeur  }
9141 {
9142   "28 = {100,200},
9143   "29 = {100,200},
9144   "2A = {100,150},
9145   "2B = {100,150},

```



```

9146     "2C = {200,300},
9147     "2D = {200,300},
9148     "2E = {   ,100},
9149     "2F = {100,   },
9150     "3F = {150,150},
9151     "5B = {   ,100},
9152     "5E = {100,100},
9153     "5F = {100,100},
9154     "80 = {   , 50},
9155     "81 = {200,250},
9156     "82 = {100,200}
9157   }
9158
9159 </eur>

```

Euler Script font (eucal).

```

9160 <+eus>
9161 \SetProtrusion
9162   [ name      = euscript ]
9163   { encoding = U,
9164     family   = eus   }
9165   {
9166     A = {100,100},
9167     B = { 50,100},
9168     C = { 50, 50},
9169     D = { 50,100},
9170     E = { 50,100},
9171     F = { 50,   },
9172     G = { 50,   },
9173     H = {   ,100},
9174     K = {   , 50},
9175     L = {   ,150},
9176     M = {   , 50},
9177     N = {   , 50},
9178     O = { 50, 50},
9179     P = { 50, 50},
9180     T = {   ,100},
9181     U = {   , 50},
9182     V = { 50, 50},
9183     W = { 50, 50},
9184     X = { 50, 50},
9185     Y = { 50,   },
9186     Z = { 50,100},
9187     "00 = {250,250},
9188     "18 = {200,200},
9189     "3A = {200,150},
9190     "40 = {   ,100},
9191     "5E = {100,100},
9192     "5F = {100,100},
9193     "66 = { 50,   },
9194     "67 = {   , 50},
9195     "6E = {200,200}
9196   }
9197
9198 \SetProtrusion
9199   [ name      = euscript-vm,
9200     load      = euscript ]
9201   { encoding = U,
9202     family   = zeus   }
9203   {
9204     "01 = {600,600},
9205     "02 = {200,200},
9206     "03 = {200,200},
9207     "04 = {200,200},
9208     "05 = {150,150},

```

```
9209 "06 = {200,200},
9210 "07 = {200,200},
9211 "08 = {100,100},
9212 "09 = {100,100},
9213 "0A = {100,100},
9214 "0B = {100,100},
9215 "0C = {100,100},
9216 "0D = {100,100},
9217 "0E = {150,150},
9218 "0F = {100,100},
9219 "10 = {150,150},
9220 "11 = {100,100},
9221 "12 = {150,100},
9222 "13 = {100,150},
9223 "14 = {150,100},
9224 "15 = {100,150},
9225 "16 = {200,100},
9226 "17 = {100,200},
9227 "19 = {150,150},
9228 "1A = {150,100},
9229 "1B = {100,150},
9230 "1C = {100,100},
9231 "1D = {100,100},
9232 "1E = {250,100},
9233 "1F = {100,250},
9234 "20 = {150,200},
9235 "21 = {150,200},
9236 "22 = {150,150},
9237 "23 = {150,150},
9238 "24 = {100,200},
9239 "25 = {150,150},
9240 "26 = {150,150},
9241 "27 = {100,100},
9242 "28 = {100,100},
9243 "29 = {100,150},
9244 "2A = {100,100},
9245 "2B = {100,100},
9246 "2C = {100,100},
9247 "2D = {150,150},
9248 "2E = {150,150},
9249 "2F = {100,100},
9250 "30 = {100,100},
9251 "31 = {100,100},
9252 "32 = {100,100},
9253 "33 = {100,100},
9254 "34 = {100,100},
9255 "35 = {100,100},
9256 "3E = {150,150},
9257 "3F = {150,150},
9258 "60 = { ,200},
9259 "61 = {200, },
9260 "62 = {100,100},
9261 "63 = {100,100},
9262 "64 = {100,100},
9263 "65 = {100,100},
9264 "68 = {300, },
9265 "69 = { ,300},
9266 "6C = {100,100},
9267 "6D = {100,100},
9268 "6F = {100,100},
9269 "72 = {100,100},
9270 "73 = {200,100},
9271 "76 = { ,100},
9272 "77 = {100, },
9273 "78 = { 50, 50},
```

```

9274     "79 = {100,100},
9275     "7A = {100,100},
9276     "7D = {150,150},
9277     "7E = {100,100},
9278     "A8 = {100,100},
9279     "A9 = {100,100},
9280     "AB = {200,200},
9281     "BA = {   ,200},
9282     "BB = {   ,200},
9283     "BD = {200,200},
9284     "DE = {200,200}
9285   }
9286
9287 </eus>

```

Euler Fraktur font (eufrak).

```

9288 <+euf>
9289 \SetProtrusion
9290   [ name      = mathfrak ]
9291   { encoding = U,
9292     family   = euf   }
9293   {
9294     A = {   , 50},
9295     B = {   , 50},
9296     C = { 50, 50},
9297     D = {   , 80},
9298     E = { 50,   },
9299     G = {   , 50},
9300     L = {   , 80},
9301     O = {   , 50},
9302     T = {   , 80},
9303     X = { 80, 50},
9304     Z = { 80, 50},
9305     b = {   , 50},
9306     c = {   , 50},
9307     k = {   , 50},
9308     p = {   , 50},
9309     q = { 50,   },
9310     v = {   , 50},
9311     w = {   , 50},
9312     x = {   , 50},
9313     1 = {100,100},
9314     2 = { 80, 80},
9315     3 = { 80, 50},
9316     4 = { 80, 50},
9317     7 = { 50, 50},
9318     "12 = {500,500},
9319     "13 = {500,500},
9320     ! = {   ,200},
9321     ' = {200,300},
9322     ( = {200,   },
9323     ) = {   ,200},
9324     * = {200,200},
9325     + = {200,250},
9326     - = {200,200},
9327     {,} = {300,300},
9328     . = {400,400},
9329     {=} = {200,200},
9330     : = {   ,200},
9331     ; = {   ,200},
9332     ] = {   ,200}
9333   }
9334
9335 </euf>
9336 </cfg-u>

```

2.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym¹⁷). The euroitc settings are hidden in the package itself (1.3.8) for ‘free software’ compliance reasons. (Not quite sure whether this is what Karl really had in mind ...)

```

9337 <*cfg-e>
9338 \SetProtrusion
9339 <zpeu> { encoding = U,
9340 <mvs> { encoding = {OT1,U},
9341 <zpeu> family = zpeu }
9342 <mvs> family = mvs }
9343 {
9344 <zpeu> E = {50, }
9345 <mvs> 164 = {50,50}, % \EUR
9346 <mvs> 068 = {50,-100} % \EURdig
9347 }
9348
9349 <*zpeu>
9350 \SetProtrusion
9351 { encoding = U,
9352 family = zpeu,
9353 shape = it* }
9354 {
9355 E = {100,-50}
9356 }
9357
9358 \SetProtrusion
9359 { encoding = U,
9360 family = {zpeus,eurosans} }
9361 {
9362 E = {100,50}
9363 }
9364
9365 \SetProtrusion
9366 { encoding = U,
9367 family = {zpeus,eurosans},
9368 shape = it* }
9369 {
9370 E = {200, }
9371 }
9372
9373 </zpeu>
9374 </cfg-e>

```

2.9 Interword spacing

Default unit is space.

```

9375 <*m-t|cmr>
9376 %%% -----
9377 %%% INTERWORD SPACING
9378
9379 </m-t|cmr>
9380 <*m-t>
9381 \SetExtraSpacing
9382 [ name = default ]
9383 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
9384 {

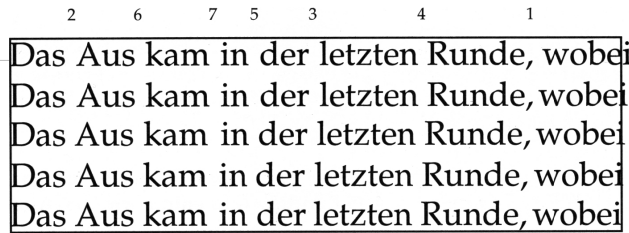
```

These settings are only a first approximation. The following reasoning is from a

17 Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

Figure 1:

Example of interword spacing (from: M. Siemoneit, *Typographisches Gestalten*, Frankfurt/M. 1989). The numbers indicate the preference for shrinking the interword space.



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

9385 {,} = { , -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)

9386 r = { , -300, 300},

- [before or] after lowercase characters with ascenders

9387 b = { , -200, 200},

9388 d = { , -200, 200},

9389 f = { , -200, 200},

9390 h = { , -200, 200},

9391 k = { , -200, 200},

9392 l = { , -200, 200},

9393 t = { , -200, 200},

- [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., ‘v’, or ‘w’

9394 c = { , -100, 100},

9395 p = { , -100, 100},

9396 v = { , -100, 100},

9397 w = { , -100, 100},

9398 z = { , -100, 100},

9399 x = { , -100, 100},

9400 y = { , -100, 100},

- [before or] after lowercase characters with x-height plus descender without additional optical space

9401 i = { , 50, -50},

9402 m = { , 50, -50},

9403 n = { , 50, -50},

9404 u = { , 50, -50},

- after colon and semicolon

9405 : = { , 200, -200},

9406 ; = { , 200, -200},

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

```
9407      . = { ,250,-250},
9408      ! = { ,250,-250},
9409      ? = { ,250,-250}
```

The order has to be reversed when enlarging is needed.’

```
9410    }
9411
9412 </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. ¹⁸

```
9413 <*cmr>
9414 \SetExtraSpacing
9415   [ name    = T2A,
9416     load    = default ]
9417   { encoding = T2A,
9418     family   = cmr }
9419   {
9420     \cyrg = { , -300, 300},
9421     \cyrb = { , -200, 200},
9422     \cyrk = { , -200, 200},
9423     \cyrs = { , -100, 100},
9424     \cyrr = { , -100, 100},
9425     \cyrh = { , -100, 100},
9426     \cyru = { , -100, 100},
9427     \cyrt = { , 50, -50},
9428     \cyrp = { , 50, -50},
9429     \cyri = { , 50, -50},
9430     \cyrishrt = { , 50, -50},
9431   }
9432
```

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:

'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.

```
9433 \SetExtraSpacing
9434   [ name      = nonfrench-cmr,
9435     load      = default,
9436     context   = nonfrench ]
9437   { encoding = {OT1,T1,LY1,OT4,QX,T5},
9438     family   = cmr }
9439   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
  \sfcode`. 3000
  \sfcode`? 3000
  \sfcode`! 3000
```

```
9440   . = {333,2000,-667},
9441   ? = {333,2000,-667},
9442   ! = {333,2000,-667},
```

```
\sfcode`: 2000
```

```
9443   : = {333,1000,-500},
```

```
\sfcode`; 1500
```

```
9444   ; = { , 500,-333},
```

```
\sfcode`\, 1250
```

```
9445   {,}= { , 250,-200}
```

```
}
```

```
9446 }
```

```
9447
```

```
9448 </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.

```
9449 (*m-t)
9450 \SetExtraSpacing
9451   [ name      = nonfrench-default,
9452     load      = default,
9453     context   = nonfrench ]
9454   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
9455   {
9456     . = {240,2000,-667},
9457     ? = {240,2000,-667},
9458     ! = {240,2000,-667},
9459     : = {240,1000,-500},
9460     ; = { , 500,-333},
9461     {,}= { , 250,-200}
9462   }
9463
```

Empty settings to prevent spurious warnings.

```
9464 \SetExtraSpacing
9465   [ name = empty ]
9466   { encoding = {TS1} }
9467   { }
9468
```

2.10 Additional kerning

Default unit is 1 em.

```
9469 %%% -----
9470 %%% ADDITIONAL KERNING
9471
```

A dummy list to be loaded when no context is active.

```
9472 \SetExtraKerning
9473   [ name = empty ]
9474   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
9475   { }
9476
```

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 1em). 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. (Wikipedia¹⁹ 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.)

```
9477 \SetExtraKerning
9478   [ name      = french-default,
9479     context   = french,
9480     unit      = space ]
9481   { encoding = {OT1,T1,LY1} }
9482   {
9483     : = {1000,}, % = \fontdimen2
9484     ; = {500,}, % - \thinspace
9485     ! = {500,},
9486     ? = {500,}
9487   }
9488
```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTeX.

```
9489 \SetExtraKerning
9490   [ name      = french-guillemets,
9491     context   = french-guillemets,
9492     load      = french-default,
9493     unit      = space ]
9494   { encoding = {T1,LY1} }
9495   {
9496     \guillemotleft = { ,800}, % = 0.8\fontdimen2
9497     \guillemotright = {800,}
9498   }
9499
```

19 https://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.


```
9500 \SetExtraKerning
9501   [ name      = french-guillemets-OT1,
9502     context   = french-guillemets,
9503     load      = french-default,
9504     unit      = space ]
9505   { encoding = OT1 }
9506   { }
9507
```

2.10.2 Turkish

```
9508 \SetExtraKerning
9509   [ name      = turkish,
9510     context   = turkish ]
9511   { encoding = {OT1,T1,LY1} }
9512   {
9513     : = {167, }, % = \thinspace
9514     ! = {167, },
9515     {=} = {167, }
9516   }
9517
9518 /m-t
9519 /config
```



```

9553     },
9554 (NewComputerModern) (l)I = {I, I, I, I, I, I, I, I, I}, % Greek
9555     J = {J},
9556     K = {K,
9557         K}, % Greek
9558     L = {L, L, L, L}, % L, L, L
9559     M = {M}, % Greek
9560     N = {N, N, N, N, N, N,
9561         N}, % Greek
9562     O = {O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O},
9563         O}, % Greek
9564 (NewComputerModern) (l)O = {O, O, O, O, O, O, O, O}, % Greek accents except O that has in-
dep. protrusion numbers (below)
9565     P = {P}, % Greek
9566 (NewComputerModern) (l)P = {P}, % Greek accents fully protruded left
9567     R = {R, R, R, R, R, R, R},
9568     S = {S, S, S, S, S, S},
9569     T = {T, T, T, T, T,
9570         T}, % Greek
9571     U = {U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U},
9572     W = {W, W, W, W},
9573     X = {X}, % Greek
9574     Y = {Y, Y, Y, Y, Y, Y},
9575 (NewComputerModern)     Y = {Y, Y, Y},
9576 (NewComputerModern) (l)Y = {Y, Y, Y, Y, Y, Y, Y, Y},
9577     Z = {Z, Z, Z, Z,
9578         Z}, % Greek
9579     a = {a, a, a, a, a, a, a, a, a, a, a, a, a, a, a, a, a, a, a, a},
9580     æ = {æ},
9581     c = {c, c, c, c, c},
9582     d = {d, d, d},
9583     e = {e, e, e, e, e, e, e, e, e, e, e, e, e, e, e, e, e, e, e, e},
9584     f = {ff}, % Unicode 64256, glyph name in Latin Modern Roman: /f_f ; in New Com-
puter Modern: /ff
9585     g = {g, g, g, g, g, g},
9586     h = {h, h, h, h, h, h},
9587     i = {i, i, i, i, i, i, i, i, i, i, i, i},
9588     j = {j},
9589     k = {k},
9590     l = {l, l, l, l}, % l, l
9591     n = {n, n, n, n, n, n},
9592     o = {o, o, o, o, o, o, o, o, o, o, o, o, o, o, o, o, o, o, o, o},
9593 (NewComputerModern)     ,o, o, o, o, o, o, o, o, o, o % Greek
9594     },
9595     r = {r, r, r, r, r, r, r},
9596     s = {s, s, s, s, s, s},
9597     t = {t, t, t, t, t}, % t
9598     u = {u, u, u, u, u, u, u, u, u, u, u, u, u, u, u, u, u, u, u, u},
9599     w = {w, w, w, w},
9600     y = {y, y, y, y, y, y, y},
9601     z = {z, z, z, z},
9602 (*NewComputerModern)
9603     α = {α, α, α, α, α, α, α, α, α, α, α, α, α, α, α, α, α, α, α, α},
9604     ε = {ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε, ε},
9605     η = {η, η, η, η, η, η, η, η, η, η, η, η, η, η, η, η, η, η, η, η},
9606     ι = {ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι, ι},
9607     ï = {ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï, ï},
9608     υ = {υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ, υ},
9609     ω = {ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω, ω},
9610 (/NewComputerModern)
9611     }
9612 (/LatinModernRoman | NewComputerModern)

```


9800 W = {Ŵ,Ŷ,Ŵ,Ŵ,Ŵ,Ŵ,
9801 W}, % Cyrillic
9802 X = {X̄,X̄,
9803 X,X,X,X}, % Cyrillic
9804 X, % Greek
9805 X}, % Roman numeral
9806 Y = {Ŷ,Ŷ,Ŷ,Ŷ,Ŷ,Ŷ,Ŷ,
9807 Y,Y}, % Cyrillic
9808 Z = {Z̄,Z̄,Z̄,Z̄,Z̄,Z̄,
9809 Z}, % Greek
9810 a = {ā,
9811 a,ā,ā}, % Cyrillic
9812 b = {b,b,b},
9813 c = {ç,ç,ç,ç,ç,
9814 c,ç}, % Cyrillic
9815 c}, % Roman numeral
9816 d = {d̄,d̄,d̄,d̄,d̄,d̄,
9817 d}, % Roman numeral
9818 e = {è,
9819 e,è,è,è}, % Cyrillic
9820 f = {f̄,f̄,/f.long,/f.DEU,/f_f},
9821 fl = {ffl,/longs_l,/longs_long_s_l,/f_l},
9822 fi = {ffi,/longs_i,/longs_long_s_i,/f_i},
9823 /f.short = {/f_f.short},
9824 g = {ḡ,ḡ,ḡ,ḡ,ḡ,ḡ,ḡ,
9825 h = {h̄,h̄,h̄,h̄,h̄,h̄,h̄,h̄,h̄,h̄,h̄,h̄,
9826 h,h}, % Cyrillic
9827 i = {ī,
9828 i,ï}, % Cyrillic
9829 i,ii,iii}, % Roman numeral
9830 j = {j,j},
9831 j}, % Cyrillic
9832 k = {k̄,k̄,k̄,k̄,k̄,k̄,
9833 l = {l̄,l̄,l̄,l̄,l̄,l̄,l̄,l̄,
9834 l}, % palochka
9835 l}, % Roman numeral
9836 m = {m̄,m̄,m̄,
9837 m}, % Roman numeral
9838 n = {ñ,n̄,n̄,n̄,n̄,n̄,n̄,n̄,n̄}, % ñ
9839 o = {ó,
9840 o,ö}, % Cyrillic
9841 p = {p̄,p̄,
9842 p,p}, % Cyrillic
9843 q = {q}, % Cyrillic
9844 r = {r̄,r̄,r̄,r̄,r̄,r̄,r̄,r̄,
9845 s = {s̄,s̄,s̄,s̄,s̄,s̄,s̄,s̄,
9846 s}, % Cyrillic
9847 t = {t̄,t̄,t̄,t̄,t̄,t̄,t̄,t̄,
9848 u = {û,
9849 v = {v̄,v̄,
9850 v}, % Roman numeral
9851 w = {ŵ,ŵ,ŵ,ŵ,ŵ,ŵ,ŵ,
9852 w}, % Cyrillic
9853 x = {x̄,x̄,
9854 x,x}, % Cyrillic
9855 x}, % Roman numeral
9856 y = {ŷ,ŷ,ŷ,ŷ,ŷ,ŷ,ŷ,ŷ,
9857 y,ÿ,ÿ,ÿ,ÿ}, % Cyrillic
9858 z = {z̄,z̄,z̄,z̄,z̄,z̄,
9859 Æ = {Æ,Æ,
9860 Æ}, % Cyrillic
9861 æ = {æ,æ,
9862 æ}, % Cyrillic
9863 DZ = {DZ̄},
9864 Dz = {Dz̄},


```

9930   ρ = {ρ,ρ̇},
9931   υ = {υ,υ̇,ϋ,υ̉,υ̊,υ̋,υ̌,υ̍,υ̎,υ̏,υ̐,υ̑,υ̒,ὐ,ὑ,υ̕,υ̖,υ̗,υ̘,υ̙,υ̚},
9932   ω = {ω,ω̇,ω̈,ω̉,ω̊,ω̋,ω̌,ω̍,ω̎,ω̏,ω̐,ω̑,ω̒,ὠ,ὡ,ω̕,ω̖,ω̗,ω̘,ω̙,ω̚},
9933 % other
9934 (i) = {(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15),(16),(17),(18),(19),(20)},
9935 (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)},
9936 [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]},
9937 ! = {!},
9938 ? = {??},
9939 . = {/onedotenleader},
9940 /endash = {/figuredash},
9941 }
9942 </EBGaramond>

```

3.1.4 Palatino

```

9943 <*Palatino>
9944 \DeclareCharacterInheritance
9945 { encoding = {TU,EU1,EU2},
9946   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.

```

9947 { A = {Ā,Ă,Ȃ,ȃ,Ȅ,ȅ,Ȇ,ȇ,Ȉ,ȉ,Ȋ,ȋ,Ȍ,ȍ,Ȏ,ȏ,Ȑ,ȑ,Ȓ,ȓ,Ȕ,ȕ,Ȗ,ȗ,Ș,ș,Ț,ț,Ȝ,ȝ,Ȟ,ȟ,Ƞ,ȡ,Ȣ,ȣ,Ȥ,ȥ,Ȧ,ȧ,Ȩ,ȩ,Ȫ,ȫ,Ȭ,ȭ,Ȯ,ȯ,Ȱ,ȱ,Ȳ,ȳ,ȴ,ȵ,ȶ,ȷ,ȸ,ȹ,Ⱥ,Ȼ,ȼ,Ƚ,Ⱦ,ȿ,Ȼ,ȼ,Ƚ,Ⱦ,ȿ},
9948   B = {Ḃ,B̈,B̉},
9949   C = {Ċ,C̈,C̉,C̊,C̋},
9950   D = {Ḋ,D̈,D̉,D̊,D̋},
9951   E = {È,É,Ê,Ë,Ē,Ē̇,Ē̈,Ē̉,Ē̊,Ē̋,Ē̌,Ē̍,Ē̎,Ē̏,Ē̐,Ē̑,Ē̒,Ē̓,Ē̔,Ē̕,Ē̖,Ē̗,Ē̘,Ē̙,Ē̚},
9952   F = {Ḟ},
9953   G = {Ġ,G̈,G̉,G̊,G̋,Ǧ,G̍,G̎,G̏},
9954   H = {Ḣ,Ḧ,H̉,H̊,H̋,Ȟ,H̍,H̎,H̏},
9955   I = {İ,Ï,Ỉ,I̊,I̋,Ǐ,I̍,I̎,Ȉ,I̐,Ȋ,I̒,I̓,I̔,I̕,I̖,I̗,I̘,I̙,I̚},
9956   J = {J̇},
9957   K = {K̇,K̈,K̉,K̊,K̋},
9958   L = {L̇,L̈,L̉,L̊,L̋,Ľ,L̍,L̎,L̏}, % L̐
9959   M = {Ṁ,M̈,M̉},
9960   N = {Ṅ,N̈,N̉,N̊,N̋,Ň,N̍,N̎,N̏},
9961   O = {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Œ,Œ̇,Œ̈,Œ̉,Œ̊,Œ̋,Œ̌,Œ̍,Œ̎,Œ̏,Œ̐,Œ̑,Œ̒,Œ̓,Œ̔,Œ̕,Œ̖,Œ̗,Œ̘,Œ̙,Œ̚},
9962   P = {Ṗ,P̈},
9963   R = {Ṙ,R̈,R̉,R̊,R̋,Ř,R̍,R̎,Ȑ},
9964   S = {Ṡ,S̈,S̉,S̊,S̋,Š,S̍,S̎,S̏},
9965   T = {Ṫ,T̈,T̉,T̊,T̋},
9966   U = {Û,Ū,Ŭ,Ŵ,Ŷ,Ÿ,Ź,Ż,Ž,Ẑ,ẑ,Ẓ,ẓ,Ẕ,ẕ,ẖ,ẗ,ẘ,ẙ,ẚ,ẛ,ẜ,ẝ,ẞ,ẟ,Ạ,ạ,Ả,ả,Ấ,ấ,Ầ,ầ,Ẩ,ẩ,Ẫ,ẽ,Ế,ế},
9967   V = {V̇,V̈},
9968   W = {Ẇ,Ẅ,W̉,W̊,W̋},
9969   X = {Ẋ,Ẍ},
9970   Y = {Ẏ,Ÿ,Ỷ,Y̊,Y̋,Y̌,Y̍,Y̎,Y̏},
9971   Z = {Ż,Z̈,Z̉,Z̊,Z̋},
9972   a = {ā,ă,ȁ,Ȃ,ȃ,Ȅ,ȅ,Ȇ,ȇ,Ȉ,ȉ,Ȋ,ȋ,Ȍ,ȍ,Ȏ,ȏ,Ȑ,ȑ,Ȓ,ȓ,Ȕ,ȕ,Ȗ,ȗ,Ș,ș,Ț,ț,Ȝ,ȝ,Ȟ,ȟ,Ƞ,ȡ,Ȣ,ȣ,Ȥ,ȥ,Ȧ,ȧ,Ȩ,ȩ,Ȫ,ȫ,Ȭ,ȭ,Ȯ,ȯ,Ȱ,ȱ,Ȳ,ȳ,ȴ,ȵ,ȶ,ȷ,ȸ,ȹ,Ⱥ,Ȼ,ȼ,Ƚ,Ⱦ,ȿ}, % a̐
9973   b = {ḃ,b̈,b̉},
9974   c = {ç,ċ,c̈,c̉,c̊,c̋},
9975   d = {ḋ,d̈,d̉,d̊,d̋},
9976   e = {è,é,ê,ë,ē,ē̇,ē̈,ē̉,ē̊,ē̋,ē̌,ē̍,ē̎,ē̏,ē̐,ē̑,ē̒,ē̓,ē̔,ē̕,ē̖,ē̗,ē̘,ē̙,ē̚},
9977   f = {ḟ,ff},
9978   g = {ġ,g̈,g̉,g̊,g̋,ǧ,g̍,g̎,g̏},
9979   h = {ḣ,ḧ,h̉,h̊,h̋,ȟ,h̍,h̎,h̏},
9980   i = {i̇,ï,ỉ,i̊,i̋,ǐ,i̍,i̎,ȉ,i̐,ȋ,i̒,i̓,i̔,i̕,i̖,i̗,i̘,i̙,i̚},
9981   j = {j̇},
9982   k = {k̇,k̈,k̉,k̊,k̋},
9983   l = {l̇,l̈,l̉,l̊,l̋}, % l̐

```

```

9984     m = {ṁ,m̈,m̉},
9985     n = {ṅ,n̈,n̉,n̊,n̋,ň,n̍,n̎}, % 'n
9986     o = {ò,ó,ô,õ,ö,ø,ō,ȯ,Ȱ,ȱ,Ȳ,ȳ,ȴ,ȵ,ȶ,ȷ,ȸ,ȹ,Ⱥ,Ȼ,ȼ,Ƚ,Ⱦ,ȿ,ȿ,ȿ,ȿ},
9987     p = {ṗ,p̈},
9988     r = {ṙ,r̈,r̉,r̊,r̋,ř,r̍,r̎},
9989     s = {ṡ,s̈,s̉,s̊,s̋,š,s̍,s̎},
9990     t = {ṫ,ẗ,t̉,t̊,t̋}, % 't
9991     u = {ù,ú,û,ü,ũ,ű,ů,ǔ,ǖ,ǘ,ǚ,ǜ,ǝ,Ǟ,ǟ,Ǡ,ǡ,Ǣ,ǣ,Ǥ,ǥ,Ǧ,ǧ,Ǩ,ǩ,Ǫ,ǫ,Ǭ,ǭ,Ǯ,ǯ,ǰ,Ǳ,ǲ,ǳ,Ǵ,ǵ},
9992     v = {v̇,v̈},
9993     w = {ẇ,ẅ,w̉,ẘ,w̋,w̌,w̍,w̎},
9994     x = {ẋ,ẍ},
9995     y = {ẏ,ÿ,ỷ,ẙ,y̋,y̌,y̍,y̎},
9996     z = {ż,z̈,z̉,z̊,z̋,ž,z̍,z̎},
9997 }
9998 /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.

```

9999 (*TU-basic
10000 \DeclareCharacterInheritance
10001   { encoding = {TU,EU1,EU2},
10002     family   = {TU-basic} }
10003   { A = {Ä,Å,Ā,Ă,Ȧ,Ȧ},
10004     a = {ä,å,ā,ă,ȧ,ȧ},
10005     C = {Ç},
10006     c = {ç},
10007     D = {Ð},
10008     E = {Ě,Ê,Ë,Ĕ},
10009     e = {ě,ê,ë,ĕ},
10010     I = {Ī,Ĭ,Ī,Ĭ},
10011     i = {ī,ĭ,ı,ı},
10012     L = {Ł},
10013     l = {ł},
10014     N = {Ñ},
10015     n = {ñ},
10016     O = {Ø,Ō,Ȫ,Ȫ,ȫ},
10017     o = {ø,ō,ȯ,Ȱ,ȱ},
10018     S = {Š},
10019     s = {š},
10020     U = {Ū,Ŭ,Ū},
10021     u = {ū,ŭ,ü},
10022     Y = {Ÿ,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.

```

10023     y = {ÿ,ß},
10024     Z = {Ž},
10025     z = {ž}
10026   }
10027 /TU-basic

```

3.1.6 Empty glyph set

Other fonts, e.g., the self-professedly awesome Font Awesome font, have no meaningful glyph arsenal at all, and should therefore be aliased so that empty settings are applied.

```

10028 (*TU-empty
10029 \DeclareCharacterInheritance

```

```

10030 { encoding = {TU,EU1,EU2},
10031   family   = {TU-empty} }
10032 { }
10033 </TU-empty>

```

3.2 Character protrusion

```

10034
10035 %%% -----
10036 %%% PROTRUSION
10037

```

3.2.1 Latin Modern Roman/New Computer Modern

```

10038 < *LatinModernRoman | NewComputerModern >
10039 \SetProtrusion
10040 < LatinModernRoman > [ name = LMR-default ]
10041 < NewComputerModern > [ name = NCM-default ]
10042 < LatinModernRoman > { encoding = {TU,EU1,EU2},
10043 < LatinModernRoman >   family   = Latin Modern Roman }
10044 < NewComputerModern > { }
10045 {
10046   A = {50,50},
10047   Æ = {50, },
10048   F = { ,50},
10049   J = {50, },
10050   K = { ,50},
10051   L = { ,50},
10052   T = {50,50},
10053   V = {50,50},
10054   W = {50,50},
10055   X = {50,50},
10056   Y = {50,50},
10057   k = { ,50},
10058   r = { ,50},
10059   t = { ,70},
10060   v = {50,50},
10061   w = {50,50},
10062   x = {50,50},
10063   y = {50,70},
10064   0 = { ,50},
10065   1 = {100,200},
10066   2 = {50,50},
10067   3 = {50,50},
10068   4 = {70,70},
10069   5 = { ,50},
10070   6 = { ,50},
10071   7 = {50,100},
10072   8 = { ,50},
10073   9 = { ,50},
10074   . = { ,700},
10075   {,} = { ,500},
10076   := { ,500},
10077   ; = { ,500},
10078   ! = { ,100},
10079   ? = { ,200},
10080   @ = {50,50},
10081   ~ = {200,250},
10082   \% = {50,50},
10083   * = {300,300},
10084   + = {250,250},
10085   - = {400,500}, % /hyphen
10086   - = {400,300}, % /endash
10087   — = {300,200}, % /emdash
10088   _ = {200,200}, % /underscore

```

```

10089 / = {200,300},
10090 /backslash = {200,300},
10091 ' = {300,400}, % /quotesingle
10092 ‘ = {300,400}, ’ = {300,400},
10093 “ = {300,300}, ” = {300,300},
10094 , = {400,400}, ,, = {400,400},
10095 ‹ = {400,400}, › = {300,500},
10096 « = {300,200}, » = {100,400},
10097 ¡ = {100, }, ¿ = {100, },
10098 ( = {300, }, ) = { ,300},
10099 < = {200,100}, > = {100,200},
10100 /braceleft = {400,200}, /braceright = {200,400},
10101 /angleleft = {400, }, /angleright = { ,400},
10102 † = {100,100},
10103 ‡ = { 80, 80},
10104 • = {200,200},
10105 · = {400,450}, % / periodcentered
10106 °C = { 80, 50},
10107 ℄ = { , 50},
10108 ° = {400,400},
10109 ™ = {100,200},
10110 © = {100,100},
10111 ® = {100,100},
10112 ª = {100,200},
10113 º = {100,200},
10114 ¹ = {200,250},
10115 º = { 50,100},
10116 ³ = { 50,100},
10117 ¬ = {200, },
10118 − = {300,300},
10119 ± = {150,200},
10120 × = {150,250},
10121 ÷ = {150,250},
10122 € = {100, },
10123 (*LatinModernRoman)
10124 /one.oldstyle = {100,100},
10125 /two.oldstyle = { 50, 50},
10126 /three.oldstyle = { 30, 80},
10127 /four.oldstyle = { 50, 50},
10128 /seven.oldstyle = { 50, 80},
10129 (/LatinModernRoman)
10130 (*NewComputerModern)
10131 A = {50,50}, % /Alphatonos
10132 Å = {120,50}, %
10133 Ǻ = {120,50}, %
10134 ǻ = {80,50}, %
10135 Ǽ = {220,50}, %
10136 ǽ = {220,50}, %
10137 ǿ = {170,50}, %
10138 Ǿ = {170,50}, %
10139 ǿ = {190,50}, %
10140 ǿ = {190,50}, %
10141 ǿ = {150,50}, %
10142 ǿ = {80,50}, %
10143 ǿ = {220,50}, %
10144 ǿ = {220,50}, %
10145 ǿ = {170,50}, %
10146 ǿ = {170,50}, %
10147 ǿ = {210,50}, %
10148 ǿ = {210,50}, %
10149 /uni1FBC.alt = {,205}, % Alpha prosgegrammeni
10150 /uni1F88.alt = {50,190}, %Alpha psili prosgegrammeni
10151 /uni1F89.alt = {,200}, %Alpha dasia prosgegrammeni
10152 /uni1F8A.alt = {130,180}, %Alpha psili baria prosgegrammeni
10153 /uni1F8B.alt = {130,190}, %Alpha dasia baria prosgegrammeni

```

10154 /uni1F8C.alt = {100,190}, %Alpha psili oxia prosgegrammeni
10155 /uni1F8D.alt = {70,190}, %Alpha dasia oxia prosgegrammeni
10156 /uni1F8E.alt = {120,190}, %Alpha psili perispomeni prosgegrammeni
10157 /uni1F8F.alt = {120,190}, %Alpha dasia perispomeni prosgegrammeni
10158 %
10159 /uni1FCC.alt = {205}, % Eta prosgegrammeni
10160 /uni1F98.alt = {185,170}, %Eta psili prosgegrammeni
10161 /uni1F99.alt = {185,170}, %Eta dasia prosgegrammeni
10162 /uni1F9A.alt = {220,170}, %Eta psili baria prosgegrammeni
10163 /uni1F9B.alt = {220,170}, %Eta dasia baria prosgegrammeni
10164 /uni1F9C.alt = {220,170}, %Eta psili oxia prosgegrammeni
10165 /uni1F9D.alt = {220,170}, %Eta dasia oxia prosgegrammeni
10166 /uni1F9E.alt = {255,170}, %Eta psili perispomeni prosgegrammeni
10167 /uni1F9F.alt = {255,170}, %Eta dasia perispomeni prosgegrammeni
10168 %
10169 O = {95,50}, %
10170 *(NewComputerModern)*
10171 $\Gamma = \{ \quad, 180 \}$, % /Gamma
10172 *(LatinModernRoman)* $\Delta = \{100,100\}$, % /Delta
10173 *(NewComputerModern)* $\Delta = \{50,50\}$, % /Delta
10174 $\Theta = \{ 50, 50 \}$, % /Theta
10175 *(LatinModernRoman)* $\Lambda = \{100,100\}$, % /Lambda
10176 *(NewComputerModern)* $\Lambda = \{50,50\}$, % /Lambda
10177 % $\Xi = \{ \}$, % /Xi
10178 % $\Pi = \{ \}$, % /Pi
10179 $\Sigma = \{ 50, 50 \}$, % /Sigma
10180 *(LatinModernRoman)* $\Upsilon = \{100,100\}$, % /Upsilon
10181 *(NewComputerModern)* $\Upsilon = \{80,80\}$, % /Upsilon
10182 $\Phi = \{ 50, 50 \}$, % /Phi
10183 $\Psi = \{ 50, 50 \}$, % /Psi
10184 *(*NewComputerModern)*
10185 $\Omega = \{ 20, 30 \}$, % /Omega
10186 $\Omega = \{150,30\}$,
10187 $\Omega = \{220,30\}$,
10188 $\Omega = \{205,30\}$,
10189 $\Omega = \{285,30\}$,
10190 $\Omega = \{285,30\}$,
10191 $\Omega = \{270,30\}$,
10192 $\Omega = \{270,30\}$,
10193 $\Omega = \{310,30\}$,
10194 $\Omega = \{310,30\}$,
10195 $\Omega = \{205,30\}$,
10196 $\Omega = \{205,30\}$,
10197 $\Omega = \{285,30\}$,
10198 $\Omega = \{285,30\}$,
10199 $\Omega = \{270,30\}$,
10200 $\Omega = \{270,30\}$,
10201 $\Omega = \{310,30\}$,
10202 $\Omega = \{310,30\}$,
10203 /uni1FFC.alt = {230}, % Omega prosgegrammeni
10204 /uni1FA8.alt = {185,190}, %Omega psili prosgegrammeni
10205 /uni1FA9.alt = {185,190}, %Omega dasia prosgegrammeni
10206 /uni1FAA.alt = {220,190}, %Omega psili baria prosgegrammeni
10207 /uni1FAB.alt = {220,190}, %Omega dasia baria prosgegrammeni
10208 /uni1FAC.alt = {220,190}, %Omega psili oxia prosgegrammeni
10209 /uni1FAD.alt = {220,190}, %Omega dasia oxia prosgegrammeni
10210 /uni1FAE.alt = {255,190}, %Omega psili perispomeni prosgegrammeni
10211 /uni1FAF.alt = {255,190}, %Omega dasia perispomeni prosgegrammeni
10212 %
10213 $\alpha = \{,50\}$,
10214 $\gamma = \{50,50\}$,
10215 $\zeta = \{,50\}$,
10216 $\vartheta = \{30,40\}$,
10217 $\iota = \{,50\}$,
10218 $\tilde{\iota} = \{-20,-30\}$,

```

10219 x = {50,50},
10220 λ = {50,50},
10221 v = {50,25},
10222 π = {50,50},
10223 σ = {,50},
10224 c = {,50},
10225 τ = {50,50},
10226 χ = {50,50},
10227 ψ = {50,50},
10228 % /uni1F98.alt = {,},

```

CMU Serif doesn't include *.end glyphs, and the OldStyle numbers' names differ.

```

10229 }
10230
10231 \SetProtrusion
10232 [ name = NCM-TU,
10233 load = NCM-default ]
10234 { encoding = {TU,EU1,EU2},
10235 family = {New Computer Modern} }
10236 {
10237 /a.end = {,330},
10238 /e.end = {,350},
10239 /k.alt = { ,50},
10240 /r.end = {,300},
10241 /m.end = {,200},
10242 /n.end = {,300},
10243 /one.oldstyle = {100,100},
10244 /two.oldstyle = { 50, 50},
10245 /three.oldstyle = { 30, 80},
10246 /four.oldstyle = { 50, 50},
10247 /seven.oldstyle = { 50, 80},
10248 }
10249
10250 \SetProtrusion
10251 [ name = CMU-TU,
10252 load = NCM-default ]
10253 { encoding = {TU,EU1,EU2},
10254 family = {CMU Serif} }
10255 {
10256 /oneoldstyle = {100,100},
10257 /twooldstyle = { 50, 50},
10258 /threeoldstyle = { 30, 80},
10259 /fouroldstyle = { 50, 50},
10260 /sevenoldstyle = { 50, 80},
10261 </NewComputerModern>
10262 }
10263
10264 \SetProtrusion
10265 <LatinModernRoman> [ name = LMR-it ]
10266 <NewComputerModern> [ name = NCM-it ]
10267 <LatinModernRoman> { encoding = {TU,EU1,EU2},
10268 <LatinModernRoman> family = Latin Modern Roman,
10269 <LatinModernRoman> shape = {it,sl} }
10270 <NewComputerModern> { }
10271 {
10272 A = {125,100},
10273 Æ = {125,-55},
10274 B = {90,-40},
10275 C = {145,-75},
10276 D = {75,-28},
10277 E = {80,-55},
10278 F = {85,-80},
10279 G = {153,-15},
10280 H = {73,-60},
10281 I = {140,-120},

```

10282 IJ = {140,-80},
10283 J = {135,-80},
10284 K = {70,-30},
10285 L = {87, 40},
10286 M = {67,-45},
10287 N = {75,-55},
10288 O = {150,-30},
10289 OE = {150,-55},
10290 P = {82,-50},
10291 Q = {150,-30},
10292 R = {75, 15},
10293 S = {90,-65},
10294 \$ = {100,-20},
10295 T = {220,-85},
10296 U = {230,-55},
10297 V = {260,-60},
10298 W = {185,-55},
10299 X = {70,-30},
10300 Y = {250,-60},
10301 Z = {90,-60},
10302 a = {150,-10},
10303 b = {170, },
10304 c = {173,-10},
10305 d = {150,-55},
10306 e = {180, },
10307 f = { , -250},
10308 g = {150,-10},
10309 h = {100, },
10310 i = {210, },
10311 ij = {210,-40},
10312 j = { , -40},
10313 k = {110,-50},
10314 l = {240,-110},
10315 m = {80, },
10316 n = {115, },
10317 o = {155, },
10318 q = {170,-40},
10319 r = {155,-40},
10320 s = {130, },
10321 t = {230,-10},
10322 u = {120, },
10323 v = {140,-25},
10324 w = {98,-20},
10325 x = {65,-40},
10326 y = {130,-20},
10327 z = {110,-80},
10328 0 = {170,-85},
10329 1 = {230,110},
10330 2 = {130,-70},
10331 3 = {140,-70},
10332 4 = {130,80},
10333 5 = {160, },
10334 6 = {175,-30},
10335 7 = {250,-150},
10336 8 = {130,-40},
10337 9 = {155,-80},
10338 . = { ,500},
10339 {,} = { ,450},
10340 : = { ,300},
10341 ; = { ,300},
10342 & = {130,30},
10343 \% = {180,50},
10344 * = {380,20},
10345 + = {180,200},
10346 @ = {180,10},

```

10347 ~ = {200,150},
10348 ( = {300, }, ) = { ,70},
10349 / = {100,100},
10350 - = {500,300}, % /hyphen
10351 – = {500,300}, % /endash
10352 — = {400,170}, % /emdash
10353 _ = {100,200}, % /underscore
10354 ' = {300,400}, % /quotesingle
10355 " = {500,300},
10356 ‘ = {800,200}, ’ = {800,-20},
10357 “ = {540,100}, ” = {500,100},
10358 , = {300,700}, ,, = {200,600},
10359 ‹ = {500,300}, › = {400,400},
10360 « = {400,100}, » = {200,300},
10361 ¡ = {200, }, ¡ = {200, },
10362 < = {300,100}, > = {200,100},
10363 /backslash = {300,300},
10364 /braceleft = {400,100}, /braceright = {200,200},
10365 † = {200, 80},
10366 ‡ = {120, 80},
10367 • = {220,100},
10368 · = {550,300}, % / periodcentered
10369 °C = {170, },
10370 ¢ = {100, 50},
10371 ¶ = {200, },
10372 ° = {500,300},
10373 ™ = {200, 70},
10374 © = { 50, 70},
10375 ® = { 50, 70},
10376 º = {140,100},
10377 º = {140,100},
10378 ¹ = {400,150},
10379 º = {250, 80},
10380 ³ = {250, 80},
10381 ¬ = {250, 80},
10382 − = {300,200},
10383 ± = {150,170},
10384 × = {200,200},
10385 ÷ = {200,200},
10386 € = {150, },
10387 (*LatinModernRoman)
10388 /one.oldstyle = {100,100},
10389 /two.oldstyle = {100, 80},
10390 /three.oldstyle = { 80, 50},
10391 /four.oldstyle = { 80, 80},
10392 /five.oldstyle = { 50, },
10393 /six.oldstyle = { 50, },
10394 /seven.oldstyle = { 80, 80},
10395 /eight.oldstyle = { 50, },
10396 (/LatinModernRoman)
10397 Γ = {100,120}, % /Gamma
10398 Δ = {120,100}, % /Delta
10399 Θ = {120, 50}, % /Theta
10400 (LatinModernRoman) Λ = {130,100}, % /Lambda
10401 (NewComputerModern) Λ = {160,100}, % /Lambda
10402 Ξ = {100,}, % /Xi
10403 Π = {100,}, % /Pi
10404 Σ = {100, 50}, % /Sigma
10405 (LatinModernRoman) Υ = {180,100}, % /Upsilon
10406 (NewComputerModern) Υ = {260,100}, % /Upsilon
10407 Φ = {130, 70}, % /Phi
10408 Ψ = {130, 50}, % /Psi
10409 Ω = { 50,}, % /Omega
10410 (*NewComputerModern)
10411 Α = {190,50}, %

```


10412 $\text{A} = \{220,50\}$, %
10413 $\text{A} = \{200,50\}$, %
10414 $\text{A} = \{300,50\}$, %
10415 $\text{A} = \{300,50\}$, %
10416 $\text{A} = \{300,50\}$, %
10417 $\text{A} = \{300,50\}$, %
10418 $\text{A} = \{320,50\}$, %
10419 $\text{A} = \{320,50\}$, %
10420 $\text{A} = \{200,50\}$, %
10421 $\text{A} = \{200,50\}$, %
10422 $\text{A} = \{300,50\}$, %
10423 $\text{A} = \{300,50\}$, %
10424 $\text{A} = \{300,50\}$, %
10425 $\text{A} = \{300,50\}$, %
10426 $\text{A} = \{320,50\}$, %
10427 $\text{A} = \{320,50\}$, %
10428 /uni1FBC.alt = {,205}, % Alpha prosgegrammeni
10429 /uni1F88.alt = {50,190}, %Alpha psili prosgegrammeni
10430 /uni1F89.alt = {,200}, %Alpha dasia prosgegrammeni
10431 /uni1F8A.alt = {130,180}, %Alpha psili baria prosgegrammeni
10432 /uni1F8B.alt = {130,190}, %Alpha dasia baria prosgegrammeni
10433 /uni1F8C.alt = {100,190}, %Alpha psili oxia prosgegrammeni
10434 /uni1F8D.alt = {70,190}, %Alpha dasia oxia prosgegrammeni
10435 /uni1F8E.alt = {120,190}, %Alpha psili perispomeni prosgegrammeni
10436 /uni1F8F.alt = {120,190}, %Alpha dasia perispomeni prosgegrammeni
10437 %
10438 /uni1FCC.alt = {,205}, % Eta prosgegrammeni
10439 /uni1F98.alt = {185,170}, %Eta psili prosgegrammeni
10440 /uni1F99.alt = {185,170}, %Eta dasia prosgegrammeni
10441 /uni1F9A.alt = {220,170}, %Eta psili baria prosgegrammeni
10442 /uni1F9B.alt = {220,170}, %Eta dasia baria prosgegrammeni
10443 /uni1F9C.alt = {220,170}, %Eta psili oxia prosgegrammeni
10444 /uni1F9D.alt = {220,170}, %Eta dasia oxia prosgegrammeni
10445 /uni1F9E.alt = {255,170}, %Eta psili perispomeni prosgegrammeni
10446 /uni1F9F.alt = {255,170}, %Eta dasia perispomeni prosgegrammeni
10447 %
10448 $\text{O} = \{95,50\}$, %
10449 $\text{\Omega} = \{120, 30\}$, % /Omega
10450 $\text{\Omega} = \{160,30\}$,
10451 $\text{\Omega} = \{250,30\}$,
10452 $\text{\Omega} = \{250,30\}$,
10453 $\text{\Omega} = \{300,30\}$,
10454 $\text{\Omega} = \{300,30\}$,
10455 $\text{\Omega} = \{300,30\}$,
10456 $\text{\Omega} = \{300,30\}$,
10457 $\text{\Omega} = \{330,30\}$,
10458 $\text{\Omega} = \{330,30\}$,
10459 $\text{\Omega} = \{30,30\}$,
10460 $\text{\Omega} = \{230,30\}$,
10461 $\text{\Omega} = \{230,30\}$,
10462 $\text{\Omega} = \{300,30\}$,
10463 $\text{\Omega} = \{300,30\}$,
10464 $\text{\Omega} = \{300,30\}$,
10465 $\text{\Omega} = \{300,30\}$,
10466 $\text{\Omega} = \{330,30\}$,
10467 $\text{\Omega} = \{330,30\}$,
10468 /uni1FFC.alt = {,230}, % Omega prosgegrammeni
10469 /uni1FA8.alt = {185,190}, %Omega psili prosgegrammeni
10470 /uni1FA9.alt = {185,190}, %Omega dasia prosgegrammeni
10471 /uni1FAA.alt = {220,190}, %Omega psili baria prosgegrammeni
10472 /uni1FAB.alt = {220,190}, %Omega dasia baria prosgegrammeni
10473 /uni1FAC.alt = {220,190}, %Omega psili oxia prosgegrammeni
10474 /uni1FAD.alt = {220,190}, %Omega dasia oxia prosgegrammeni
10475 /uni1FAE.alt = {255,190}, %Omega psili perispomeni prosgegrammeni
10476 /uni1FAF.alt = {255,190}, %Omega dasia perispomeni prosgegrammeni

```

10477 %
10478   α = {50,50},
10479   γ = {100,50},
10480   δ = {30,50},
10481   ε = {30,},
10482   ζ = {20,50},
10483   ϑ = {30,40},
10484   ι = {50,},
10485   ï = {-20,-30},
10486   κ = {50,50},
10487   λ = {-20,50},
10488   ν = {50,25},
10489   ο = {40,},
10490   π = {50,50},
10491   σ = {40,50},
10492   ς = {20,50},
10493   τ = {50,50},
10494   υ = {80,},
10495   φ = {80,},
10496   χ = {20,},
10497   ψ = {80,},
10498 % /uni1F98.alt = {,},
10499 }
10500
10501 \SetProtrusion
10502 [ name = NCM-it-TU,
10503   load = NCM-it ]
10504 { encoding = {TU,EU1,EU2},
10505   family = {New Computer Modern},
10506   shape = {it,s1} }
10507 {
10508   /a.end = {,330}, %Fix
10509   /e.end = {,350}, %Fix
10510   /k.alt = { ,50}, %Fix
10511   /r.end = {,300}, %Fix
10512   /m.end = {,200}, %Fix
10513   /n.end = {,300}, %Fix
10514   /one.oldstyle = {100,100},
10515   /two.oldstyle = {100, 80},
10516   /three.oldstyle = { 80, 50},
10517   /four.oldstyle = { 80, 80},
10518   /five.oldstyle = { 50,  },
10519   /six.oldstyle = { 50,  },
10520   /seven.oldstyle = { 80, 80},
10521   /eight.oldstyle = { 50,  },
10522 }
10523
10524 \SetProtrusion
10525 [ name = CMU-it-TU,
10526   load = NCM-it ]
10527 { encoding = {TU,EU1,EU2},
10528   family = {CMU Serif},
10529   shape = {it,s1} }
10530 {
10531   /oneoldstyle = {100,100},
10532   /twooldstyle = {100, 80},
10533   /threeoldstyle = { 80, 50},
10534   /fouroldstyle = { 80, 80},
10535   /fiveoldstyle = { 50,  },
10536   /sixoldstyle = { 50,  },
10537   /sevenoldstyle = { 80, 80},
10538   /eightoldstyle = { 50,  },
10539 /NewComputerModern
10540 }
10541 /LatinModernRoman|NewComputerModern

```

3.2.2 Charis SIL

```

10542 (*CharisSIL)
10543 \SetProtrusion
10544 [ name = Charis-default ]
10545 { encoding = {TU,EU1,EU2},
10546   family = Charis SIL }
10547 {
10548   A = {50,50},
10549   Æ = {50,50},
10550   C = {50, },
10551   D = { ,50},
10552   F = { ,50},
10553   G = {50, },
10554   J = {100, },
10555   K = { ,50},
10556   L = { ,50},
10557   Ḷ = { ,100},
10558   O = {50,50},
10559   Œ = {50, },
10560   P = { ,50},
10561   Q = {50,70},
10562   R = { ,50},
10563   ß = { ,40}, % capital sharp s
10564   T = {50,50},
10565   V = {50,50},
10566   W = {50,50},
10567   X = {50,50},
10568   Y = {50,50},
10569   k = { ,50},
10570   ḷ = { ,150},
10571   r = { ,50},
10572   t = { ,50},
10573   v = {50,50},
10574   w = {50,50},
10575   x = {50,50},
10576   y = { ,50},
10577   1 = {150,150},
10578   2 = {50,50},
10579   3 = {50, },
10580   4 = {100,50},
10581   6 = {50, },
10582   7 = {50,80},
10583   9 = {50,50},
10584   . = { ,600},
10585   {,} = { ,500},
10586   : = { ,400},
10587   ; = { ,300},
10588   ! = { ,100},
10589   ? = { ,200},
10590   @ = {50,50},
10591   ~ = {200,250},
10592   \% = { ,50},
10593   * = {300,300},
10594   + = {200,250},
10595   / = { ,200},
10596   /backslash = {150,200},
10597   | = {200,200},
10598   - = {400,500}, % hyphen
10599   - = {200,300}, % endash
10600   — = {150,250}, % emdash
10601   — = {200,200}, % Horizontal Bar = \texttwelveudash
10602   - = {150,150}, % Figure Dash = \textthreequartersemdash
10603   _ = {100,100},
10604   {=} = {100,100},

```

10605 ‘ = {300,400}, ’ = {300,400},
10606 “ = {300,300}, ” = {300,300},
10607 , = {400,400}, „ = {300,300},
10608 ‹ = {400,300}, › = {300,400},
10609 « = {200,200}, » = {150,300},
10610 ¡ = {100, }, ¿ = {100, },
10611 (= {200, },) = { ,200},
10612 < = {200,150}, > = {100,200},
10613 [= {100, },] = { ,100},
10614 /braceleft = {200, }, /braceright = { ,300},
10615 † = { 80, 80},
10616 ‡ = {100,100},
10617 • = {200,200},
10618 ° = {150,200},
10619 ™ = {150,150},
10620 ¢ = { 50, },
10621 £ = { 50, },
10622 † = {200,200},
10623 © = {100,100},
10624 ® = {100,100},
10625 º = {100,200},
10626 º = {200,200},
10627 ¬ = {200, 50},
10628 µ = { ,100},
10629 ¶ = { ,100},
10630 · = {300,400},
10631 ¹ = {200,300},
10632 ² = {100,200},
10633 ³ = {100,200},
10634 € = {100, },
10635 ± = {150,200},
10636 × = {200,200},
10637 ÷ = {250,250},
10638 /minus = {200,200},
10639 − = {200,200},
10640 % Cyrillic
10641 Б = { ,50},
10642 Г = { ,130},
10643 Ж = {50,50},
10644 З = {30,50},
10645 Л = {50, },
10646 У = {50,50},
10647 Ф = {50,50},
10648 Ч = {100, },
10649 Ъ = { ,50},
10650 б = { ,50},
10651 Э = {50,50},
10652 Ю = { ,40},
10653 Я = {50, },
10654 В = {50,50},
10655 € = {50, },
10656 Ъ = {50,100},
10657 € = {50, },
10658 Ъ = {50,50},
10659 Ъ = { ,50},
10660 Ъ = {50,50},
10661 Ъ = {100,100},
10662 Ъ = {50,50},
10663 Ъ = { ,50},
10664 б = { ,50},
10665 Ъ = {50,80},
10666 Ъ = { ,80},
10667 Ъ = {50,50},
10668 Ј = {50, },
10669 ЈХ = {50,40},

```

10670   K = { ,50},
10671   Я = {50, },
10672   Л̣ = { ,50},
10673   Ѓ = { ,50},
10674   đ = { ,100},
10675   б = {50,50},
10676   г = { ,70},
10677   к = { ,50},
10678   л = {50, },
10679   т = {50,50},
10680   ф = {50,50},
10681   ч = {50, },
10682   ъ = { ,50},
10683   ь = { ,50},
10684   э = { ,50},
10685   я = {50, },
10686   љ = {50, },
10687   њ = { ,50},
10688   џ = { ,50},
10689   v = {50,50},
10690   е = {50, },
10691   њ = { ,50},
10692   y = {50,50},
10693   Ѓ = { ,50},
10694   Ѓ = { ,50},
10695   đ = { ,100},
10696   ʒ = {100,100},
10697   ʒ = {50,50},
10698   љ = {50,70},
10699   џ = { ,70},
10700   яе = {50,30},
10701   Л̣ = { ,50},
10702   Ѓ = { ,50},
10703   %   Д П Ц Ш Щ Ы Ъ Ь Ѡ ѡ Т Ц Ъ Э Э д
10704   %   в д ж з и м н п ц ш ы ю ѧ е ѧ ѡ э э ѡ ц ʒ d e џ л ж р
10705   % Greek
10706   Δ = {50,50},
10707   Ψ = {50,50},
10708   γ = {70,70},
10709   λ = {40,70},
10710   π = {40,50},
10711   ρ = { ,50},
10712   σ = { ,50},
10713   χ = {50,50},
10714 }
10715
10716 \SetProtrusion
10717 [ name = Charis-it ]
10718 { encoding = {TU,EU1,EU2},
10719   family = Charis SIL,
10720   shape = {it,s1} }
10721 {
10722   C = {50, },
10723   G = {50, },
10724   J = {50, },
10725   L = {50,50},
10726   O = {50, },
10727   Œ = {50, },
10728   Q = {50, },
10729   S = {50, },
10730   $ = {50, },
10731   T = {70, },
10732   o = {50,50},
10733   p = { ,50},
10734   q = {50, },

```

```

10735     t = { ,50},
10736     w = { ,50},
10737     y = { ,50},
10738     1 = {150,100},
10739     3 = {50, },
10740     4 = {100, },
10741     6 = {50, },
10742     7 = {100, },
10743     . = { ,700},
10744     {,}= { ,600},
10745     : = { ,400},
10746     ; = { ,400},
10747     ? = { ,150},
10748     & = { ,80},
10749     \% = {50,50},
10750     * = {300,200},
10751     + = {250,250},
10752     @ = {80,50},
10753     ~ = {150,150},
10754     / = { ,150},
10755     /backslash = {150,150},
10756     - = {300,400}, % hyphen
10757     - = {200,300}, % endash
10758     — = {150,200}, % emdash
10759     _ = { ,100},
10760     {=} = {200,200},
10761     ± = {150,200},
10762     × = {250,250},
10763     ÷ = {250,250},
10764     ° = {150,200},
10765     · = {300,400},
10766     ‘ = {400,200}, ’ = {400,200},
10767     “ = {300,200}, ” = {400,200},
10768     , = {200,500}, „ = {150,500},
10769     ‹ = {300,400}, › = {200,500},
10770     « = {200,300}, » = {150,400},
10771     ( = {200, }, ) = { ,200},
10772     < = {200,200}, > = {200,200},
10773     /braceleft = {300, }, /braceright = { ,200},
10774     % Cyrillic
10775     Ж = {50,30},
10776     Л = {50, },
10777     У = {50,30},
10778     Ф = {50, },
10779     Ч = {100, },
10780     Ъ = { ,50},
10781     Ь = { ,50},
10782     Э = {50,50},
10783     Я = {50, },
10784     В = {50,50},
10785     Љ = {50,50},
10786     Ђ = {140,100},
10787     Ѓ = {70,50},
10788     Ћ = {50,80},
10789     Ў = { ,80},
10790     Ђ = {50,50},
10791     г = {50,50},
10792     д = {50,30},
10793     м = {50, },
10794     ф = {50, },
10795     ч = {50, },
10796     ъ = { ,50},
10797     ь = { ,50},
10798     э = { ,50},
10799     я = {50, },

```

```

10800     њ = {50,50},
10801     њ = { ,50},
10802     v = {50,50},
10803     Ъ = { ,50},
10804     ѳ = {140,100},
10805     ϳ = {70,50},
10806     Ѵ = {50,70},
10807     ѵ = { ,70},
10808     % Greek
10809     Γ = { ,130},
10810     Δ = {50,50},
10811     Ψ = {50,50},
10812     γ = {70,70},
10813     λ = {40,70},
10814     π = {40,50},
10815     ρ = { ,50},
10816     σ = { ,50},
10817     χ = {50,50},
10818 }

```

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 XeTeX (where we check for glyph name).

```

10819
10820 % quick and dirty -- maybe we'll promote this to a
10821 % regular key some time
10822 \define@key{MT@pr@c}{command}{\csname #1\endcsname}
10823
10824 % glyph names have changed with version 5.0 of Charis SIL:
10825 % before: /a.SC, /b.SC, ...
10826 % after: /a.sc, /b.sc, ...
10827 \ifx\MT@lua\undefined
10828   \gdef\MT@get@CHARIS@SC{
10829     % test whether glyph "a.sc" exists
10830     \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
10831       \gdef\MT@CHARIS@SC{sc}%
10832     \else
10833       \gdef\MT@CHARIS@SC{SC}%
10834     \fi
10835   }
10836 \else
10837   \gdef\MT@get@CHARIS@SC{
10838     \gdef\MT@CHARIS@SC{\MT@lua{
10839       % check font version
10840       % -- why doesn't this work?:
10841       % f = font.getfont(font.current());
10842       % i = fontloader.info(f.filename);
10843       % if (tonumber(i.version) < 5) then;
10844       % if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
10845       %   tex.print("SC");
10846       % else;
10847       %   tex.print("sc");
10848       % end
10849     }}
10850   }
10851 \fi
10852
10853 \SetProtrusion
10854   [ name      = Charis-sc,
10855     load      = Charis-default,
10856     command   = {MT@get@CHARIS@SC} ]
10857   { encoding = {TU,EU1,EU2},
10858     family   = Charis SIL,
10859     shape    = {sc} }

```

```

10860 {
10861 %   A = {100,100}, % etc., doesn't work with \textsc
10862   /a.\MT@CHARIS@SC = {100,100},
10863   /c.\MT@CHARIS@SC = {50, },
10864   /d.\MT@CHARIS@SC = { ,50},
10865   /f.\MT@CHARIS@SC = { ,50},
10866   /g.\MT@CHARIS@SC = {50, },
10867   /j.\MT@CHARIS@SC = {100, },
10868   /k.\MT@CHARIS@SC = { ,50},
10869   /l.\MT@CHARIS@SC = { ,50},
10870 /f_l.\MT@CHARIS@SC = { ,50},
10871   /o.\MT@CHARIS@SC = {50,50},
10872   /oe.\MT@CHARIS@SC = {50, },
10873   /q.\MT@CHARIS@SC = {50,70},
10874   /r.\MT@CHARIS@SC = { ,50},
10875   /t.\MT@CHARIS@SC = {50,100},
10876   /v.\MT@CHARIS@SC = {50,50},
10877   /w.\MT@CHARIS@SC = {50,50},
10878   /x.\MT@CHARIS@SC = {50,50},
10879   /y.\MT@CHARIS@SC = {50,50}
10880 }
10881 (<CharisSIL)

```

3.2.3 EB Garamond

```

10882 (<*EBGaramond)
10883 \SetProtrusion
10884 [ name      = EBGaramond-TU,
10885   load      = EBGaramond-T1-LF ]
10886 { encoding = {TU,EU1,EU2},
10887   family   = EBGaramond }
10888 {
10889   /one.tosf = {150,150},
10890   /two.tosf = {50,50},
10891   /three.tosf = {50,50},
10892   /four.tosf = {50,50},
10893   /five.tosf = {50,50},
10894   /six.tosf = {50,50},
10895   /seven.tosf = {50,80},
10896   /eight.tosf = {50,50},
10897   /nine.tosf = {50,50},
10898   /one.lf = {50,50},
10899   /two.lf = {50,50},
10900   /four.lf = {50,50},
10901   /seven.lf = {50,50},
10902   /one.osf = {50,50},
10903   /two.osf = {50,50},
10904   /four.osf = {50,50},
10905   /seven.osf = {50,50},
10906   IV = { , 35},
10907   VI = { 35, },
10908   VII = { 30, },
10909   VIII = { 25, },
10910   IX = { , 35},
10911   XI = { 35, },
10912   XII = { 30, },
10913   iv = { , 25},
10914   vi = { 25, },
10915   vii = { 20, },
10916   viii = { 20, },
10917   ix = { , 25},
10918   xi = { 25, },
10919   xii = { 20, },
10920 % textcomp
10921 \textquotesingle = {400,500},
10922 _ = {200,250},

```


10923 $f = \{ \quad, 100\}$,
10924 $\text{Œ} = \{ 50, \quad\}$,
10925 $\dagger = \{100, 100\}$,
10926 $\ddagger = \{ 80, 80\}$,
10927 $\bullet = \{ \quad, 100\}$,
10928 $\cdot = \{300, 400\}$, % periodcentered
10929 $/\text{twodotenleader} = \{150, 200\}$,
10930 $/\text{ellipsis} = \{100, 150\}$,
10931 $^{\circ}\text{C} = \{ 80, \quad\}$,
10932 $^{\circ} = \{400, 400\}$,
10933 $^{\text{TM}} = \{100, 100\}$,
10934 $^{\text{C}} = \{100, 100\}$,
10935 $^{\text{R}} = \{100, 100\}$,
10936 $^{\text{S}} = \{200, 200\}$,
10937 $^{\text{E}} = \{200, 200\}$,
10938 $^{\text{I}} = \{200, 200\}$,
10939 $^{\text{2}} = \{200, 200\}$,
10940 $^{\text{3}} = \{200, 200\}$,
10941 $\neg = \{200, \quad\}$,
10942 $\text{¶} = \{ \quad, 100\}$,
10943 $- = \{300, 300\}$, % minus
10944 $\pm = \{150, 200\}$,
10945 $\times = \{100, 150\}$,
10946 $\div = \{150, 200\}$,
10947 $\text{€} = \{ 50, 100\}$,
10948 $\text{¥} = \{ 50, 50\}$,
10949 % Greek
10950 $\Gamma = \{ \quad, 150\}$,
10951 $\Delta = \{100, 100\}$,
10952 $\Theta = \{ 50, 50\}$,
10953 $\Lambda = \{100, 100\}$,
10954 $\Xi = \{ 50, 50\}$,
10955 $\Upsilon = \{100, 100\}$,
10956 $\Phi = \{ 50, 50\}$,
10957 $\Psi = \{ 50, 50\}$,
10958 $\Omega = \{ \quad, 50\}$,
10959 $\zeta = \{ \quad, 50\}$,
10960 $\lambda = \{ 50, 50\}$,
10961 $\gamma = \{ 50, 50\}$,
10962 $\pi = \{ 50, 50\}$,
10963 $\rho = \{ \quad, 50\}$,
10964 $\sigma = \{ 50, 50\}$,
10965 $\tau = \{ 50, 50\}$,
10966 $\chi = \{ 50, 50\}$,
10967 $\phi = \{ 50, 50\}$,
10968 $\varphi = \{ 50, 50\}$,
10969 $\psi = \{ 50, 50\}$,
10970 % Cyrillic
10971 $\Gamma = \{ \quad, 150\}$,
10972 $\text{Д} = \{ 50, 50\}$,
10973 $\text{Ж} = \{ 50, 50\}$,
10974 $\text{К} = \{ \quad, 50\}$,
10975 $\text{Л} = \{ 50, \quad\}$,
10976 $\text{Љ} = \{ 50, 50\}$,
10977 $\text{З} = \{ 50, 50\}$,
10978 $\text{У} = \{ 50, 100\}$,
10979 $\text{Ф} = \{ 50, 50\}$,
10980 $\text{Ч} = \{ 70, \quad\}$,
10981 $\text{Я} = \{ 50, \quad\}$,
10982 $\text{Ђ} = \{ 50, 50\}$,
10983 $\text{Б} = \{ \quad, 50\}$,
10984 $\text{ж} = \{ 50, 50\}$,
10985 $\text{ф} = \{ 50, 50\}$,
10986 $\text{џ} = \{ 50, 50\}$,
10987 $\text{Ѣ} = \{ 50, 50\}$,

```

10988   r = {   , 50},
10989   V = { 50, 50},
10990   % other
10991   Þ = {   , 50},
10992   þ = {   , 50},
10993   Λ = {100,100},
10994   (i) = { 35, 65},
10995   (a) = { 30, 60},
10996   }
10997
10998 \SetProtrusion
10999 [ name      = EBGaramond-it-TU,
11000   load      = EBGaramond-it-T1-LF ]
11001 { encoding = {TU,EU1,EU2},
11002   family   = EBGaramond,
11003   shape    = it }
11004 {
11005   /zero.tosf = {150,150},
11006   /one.tosf  = {150,150},
11007   /two.tosf  = {80,80},
11008   /three.tosf = {50,80},
11009   /four.tosf = {50,80},
11010   /five.tosf = {50,80},
11011   /six.tosf  = {50,50},
11012   /seven.tosf = {50,100},
11013   /eight.tosf = {50,50},
11014   /nine.tosf = {50,80},
11015   /one.lf    = {50,50},
11016   /two.lf    = {50,50},
11017   /three.lf  = {80,50},
11018   /four.lf   = {50,50},
11019   /five.lf   = {50,50},
11020   /six.lf    = {50,50},
11021   /seven.lf  = {50,50},
11022   /eight.lf  = {50,50},
11023   /nine.lf   = {50,  },
11024   /one.osf   = {50,50},
11025   /two.osf   = {50,50},
11026   /three.osf = {  ,80},
11027   /four.osf  = {50,50},
11028   /seven.osf = {50,50},
11029   % textcomp
11030   \textquotesingle = {800,100},
11031   - = {300,300}, % minus
11032   ¸ = {200,250},
11033   † = {200,100},
11034   ‡ = { 80, 80},
11035   • = {300,  },
11036   °C = {200,  },
11037   f = {100,  },
11038   ℄ = {100,  },
11039   ™ = {200,  },
11040   © = {200,100},
11041   ® = {200,100},
11042   ¬ = {300,  },
11043   ° = {500,100},
11044   ± = {200,150},
11045   ¹ = {300,100},
11046   ² = {300,  },
11047   ³ = {300,  },
11048   · = {300,500}, % periodcentered
11049   /twodotenleader = {150,300},
11050   /ellipsis = {100,200},
11051   € = {100,  },
11052   × = {200,100},

```

```

11053 ÷ = {200,200},
11054 ¶ = { ,100},
11055 ª = {200,200},
11056 º = {200,200},
11057 ¥ = { 50, 50},
11058 % Greek
11059 Δ = {150, },
11060 Θ = { 50, },
11061 Λ = {150, },
11062 Υ = {100, 50},
11063 Φ = { 50, },
11064 Χ = { 50, },
11065 Ψ = {100, },
11066 Ω = { 50, },
11067 γ = { , 50},
11068 λ = { 50, },
11069 % Cyrillic
11070 Я = { 50, },
11071 Ч = {100, },
11072 З = {100, },
11073 % other
11074 Ъ = { 50, 50},
11075 ъ = { , 50},
11076 }
11077
11078 \SetProtrusion
11079 [ name = EBGaramond-sc-TU,
11080 load = EBGaramond-TU ]
11081 { encoding = {TU,EU1,EU2},
11082 family = EBGaramond,
11083 shape = sc }
11084 {
11085 a = {50,50},
11086 \ae = {50, },
11087 d = { ,50},
11088 f = { ,50},
11089 g = {50, },
11090 j = {50, },
11091 l = { ,50},
11092 o = {50,50},
11093 \oe = {50, },
11094 q = {50,70},
11095 r = { , 0},
11096 t = {50,50},
11097 y = {50,50},
11098 % Greek
11099 α = {50,50},
11100 γ = { ,50},
11101 δ = {50,50},
11102 λ = {50,50},
11103 ο = {50,50},
11104 τ = {50,50},
11105 υ = {50,50},
11106 ψ = {50,50},
11107 % Cyrillic
11108 τ = {50,50},
11109 }
11110
11111 \SetProtrusion
11112 [ name = EBGaramond-scit-TU,
11113 load = EBGaramond-it-TU ]
11114 { encoding = {TU,EU1,EU2},
11115 family = EBGaramond,
11116 shape = scit }

```

```

11117 {
11118   a = {50,50},
11119   \ae = {50, },
11120   d = { ,50},
11121   f = { ,50},
11122   g = {50, },
11123   j = {50, },
11124   l = { ,50},
11125   o = {50,50},
11126   \oe = {50, },
11127   q = {50,70},
11128   r = { , 0},
11129   t = {50,50},
11130   y = {50,50},
11131 % Greek
11132   α = {50,50},
11133   γ = { ,50},
11134   δ = {50,50},
11135   λ = {50,50},
11136   ο = {50,50},
11137   τ = {50,50},
11138   υ = {50,50},
11139   ψ = {50,50},
11140 % Cyrillic
11141   τ = {50,50},
11142 }
11143 </EBGaramond>

```

3.2.4 Palatino

```

11144 <*Palatino>
11145 \SetProtrusion
11146 [ name = palatino-default ]
11147 { encoding = {TU,EU1,EU2},
11148   family = {Palatino} }
11149 {
11150   A = {50,50},
11151   D = { ,50},
11152   J = {50, },
11153   K = { ,50},
11154   L = { ,50},
11155   O = {25, },
11156   T = {50,50},
11157   V = {50,50},
11158   W = {50,50},
11159   X = {50,50},
11160   Y = {50,50},
11161   b = { ,25},
11162   d = {25,30},
11163   f = { ,50},
11164   g = { ,100},
11165   k = { ,50},
11166   p = { ,50},
11167   q = {50, },
11168   r = { ,50},
11169   t = { ,50}, ◆ = { ,50}, ◆ = { ,50},
11170   v = {75,50},
11171   w = {50,50},
11172   x = {50,50},
11173   y = {50,70},
11174   1 = {100,50},
11175   2 = {25,50},
11176   4 = {50, },
11177   6 = {50, },
11178   9 = {25, },

```

```

11179   Æ = {100, },
11180   Ē = {25, },
11181   . = { ,700},      .. = { ,350},      ... = { ,150},
11182   {,} = { ,500},
11183   : = { ,500},
11184   ; = { ,500},
11185   ! = { ,100},      !! = { ,100},
11186   ? = { ,200},      ʔ = { ,200},
11187   @ = {50,50},
11188   ~ = {200,250},
11189   & = {50,100},
11190   \% = {100,100},
11191   * = {200,200},
11192   + = {250,250},
11193   ( = {100, },      ) = { ,300},
11194   / = {200,300},
11195   - = {400,500},
11196   \textendash      = {300,300},      \textemdash      = {200,200},
11197   \textquoteleft   = {500,700},      \textquoteright   = {500,700},
11198   \textquotedblleft = {300,400},      \textquotedblright = {300,400},
11199   \textbackslash     = {200,300},
11200   \quotesinglbase   = {400,400},      \quotedblbase     = {400,400},
11201   \guilsinglleft    = {400,400},      \guilsinglright   = {300,500},
11202   \guillemotleft   = {300,300},      \guillemotright   = {200,400},
11203   \textexclamdown   = {100, },      \textquestiondown = {100, },
11204   \textbraceleft    = {400,200},      \textbraceright   = {200,400},
11205   \textless         = {200,100},      \textgreater       = {100,200},
11206   ≤                 = {200,100},      ≥                 = {100,200},
11207   \textminus        = {300,300},
11208   \texttrademark    = {200,200},
11209   \textcopyright    = {200,200},
11210   \textregistered   = {200,200},
11211   \textdegree       = {300,300},
11212   ¡                 = {450,500},      ¬                 = {250,150},
11213   ¯                 = {150,250},
11214   ·                 = {850, 700},
11215   ¶                 = {100,0},
11216   ×                 = {150, 300},
11217   ª                 = {300,300},      º                 = {300,300},
11218   ⁰ = {200,400},
11219   ¹ = {400,350},      ² = {200,300},      ³ = {250,400},
11220   ⁴ = {250,350},      ⁵ = {200,300},      ⁶ = {250,400},
11221   ⁷ = {200,450},      ⁸ = {250,400},      ⁹ = {200,350},
11222   ⁰ = {200,400},
11223   ¹ = {400,250},      ² = {200,300},      ³ = {250,400},
11224   ⁴ = {250,350},      ⁵ = {200,300},      ⁶ = {250,400},
11225   ⁷ = {200,450},      ⁸ = {250,400},      ⁹ = {200,350},
11226   ± = {150,100},      ÷ = {300,300},
11227   þ = { ,25},
11228   ˙ = {300,450},      ˘ = {300,450},
11229   ˚ = {300,450},      ˛ = {300,450},
11230   † = {200,250},      ‡ = {200,250},
11231   π = {50, },
11232   f = { ,50},
11233   № = {100,150},
11234   \textservicemark = {100,200},
11235   - = {400,500},      - = {400,500},      - = {200,300},
11236   - = {205,305},      - = {200,300},      - = {50,150},
11237   ● = {125,200},
11238   % /a.sc = {50,50},
11239 }
11240
11241 \SetProtrusion
11242 [ name = palatino-it ]
11243 { encoding = {TU,EU1,EU2},

```

```

11244     family = {Palatino},
11245     shape   = {it,sl} }
11246     {
11247         A = {50,50},
11248         Æ = {50, },
11249         B = {50, },
11250         C = {50, },
11251         D = {50,50},
11252         E = {50, },
11253         F = {50, },
11254         G = {50, },
11255         H = {50, },
11256         K = {50, },
11257         L = {50, },
11258         O = {50, },
11259         Œ = {50, },
11260         P = {50, },
11261         Q = {50, },
11262         R = {50, },
11263         S = {50, },
11264         $ = {50, },
11265         T = {100, },
11266         U = {50, },
11267         V = {100,50},
11268         W = {50, },
11269         X = {50, },
11270         Y = {100,50},
11271         b = { ,50},
11272         c = {25, },
11273         g = {75, },
11274         i = {25, },
11275         m = { ,50},
11276         n = { ,50},
11277         p = { ,25},
11278         q = {25, },
11279         x = { ,50},
11280         1 = {100, },
11281         2 = {50, },
11282         4 = {50, },
11283         7 = {50, },
11284         . = { ,500},      .. = { ,350},      ... = { ,200},
11285         {,} = { ,500},
11286         : = { ,300},
11287         ; = { ,300},
11288         ? = { ,300},      ? = { ,300},
11289         & = {50,50},
11290         \% = {100,100},
11291         * = {200,200},
11292         + = {150,200},
11293         @ = {50,50},
11294         ~ = {200,150},
11295         ( = {200, },      ) = { ,200},
11296         / = {100,200},
11297         - = {300,500},
11298         \textendash      = {300,300},      \textemdash      = {200,200},
11299         \textquoteleft  = {700,400},      \textquoteright  = {700,400},
11300         \textquotedblleft = {500,300},      \textquotedblright = {500,300},
11301         _ = {100,100},
11302         \textbackslash    = {100,200},
11303         \quotesinglbase   = {500,500},      \quotedblbase    = {400,400},
11304         \guilsinglleft    = {400,400},      \guilsinglright  = {300,500},
11305         \guillemotleft    = {300,300},      \guillemotright  = {300,300},
11306         \textexclamdown   = {100, },      \textquestiondown = {200, },
11307         \textbraceleft    = {200,100},      \textbraceright  = {200,200},
11308         \textless        = {300,100},      \textgreater     = {200,100},

```

```

11309 ≤ = {200,100}, ≥ = {100,200},
11310 | = {450,500}, ¬ = {250,150},
11311 · = {850, 700},
11312 ¶ = {100,0},
11313 × = {150, 300},
11314 ° = {300,300}, ° = {300,250},
11315 ° = {300,200},
11316 ¹ = {300,150}, ² = {350,200}, ³ = {250,150},
11317 ⁴ = {350,100}, ⁵ = {300, 50}, ⁶ = {400,100},
11318 ⁷ = {400, 50}, ⁸ = {250, 50}, ⁹ = {300, 50},
11319 ⁰ = {300,300},
11320 ¹ = {300,350}, ² = {300,150}, ³ = {250,250},
11321 ⁴ = {400,200}, ⁵ = {300,100}, ⁶ = {450,200},
11322 ⁷ = {450,150}, ⁸ = {400,250}, ⁹ = {400,200},
11323 ± = {150,100}, ÷ = {300,300},
11324 þ = { 50, },
11325 † = {250,200}, ‡ = {250,200},
11326 ⁙ = {300,450}, ⁚ = {300,450},
11327 ⁛ = {300,450}, ⁜ = {300,450},
11328 - = {300,500}, - = {300,500}, - = {100,300},
11329 - = {125,305}, - = {200,300}, - = {125,150},
11330 • = {125,200}

11331 }
11332
11333 \SetProtrusion
11334 [ name = palatino-sc,
11335 load = palatino-default ]
11336 { encoding = {TU,EU1,EU2},
11337 family = {Palatino},
11338 shape = sc }
11339 {

11340 a = {50,50},
11341 æ = {50, },
11342 b = { 0, 0},
11343 d = { 0, 0},
11344 f = { 0, 0},
11345 g = { 0, 0},
11346 j = {50, },
11347 l = { ,50},
11348 o = { 0, 0},
11349 p = { 0, 0},
11350 q = { 0, },
11351 r = { , 0},
11352 t = {50,50},
11353 y = {50,50},
11354 fl = { 0,50},
11355 ffl = { 0,50},
11356 ◊ = { 0,50},
11357 ◊ = { 0,50}

11358 }
11359 </Palatino>

```

3.2.5 Basic glyph set

The protrusion settings will still be loaded from `microtype.cfg`.

```
11360 <TU-basic> %% No settings.
```

3.2.6 Empty glyph set

```

11361 <*TU-empty>
11362 \SetProtrusion
11363 [ name = empty ]
11364 { encoding = {TU,EU1,EU2},

```

```
11365     family = {TU-empty} }
11366     { }
11367 (</TU-empty>
11368
```


4 Auxiliary file for micro fine tuning

This file may be used to test protrusion and (less so) expansion settings.

```

11369 (*test)
11370 \documentclass{article}
11371 %% options are passed through to microtype
11372 \usepackage[stretch=50]{microtype-show}
11373
11374 %% options for microtype-show
11375 \ShowGlyphIndextrue
11376 \ShowMissingGlyphstrue
11377 \def\GlyphScaleFactor{2}
11378
11379 %% load any required font packages:
11380 \ifpdftex
11381 \usepackage[T1]{fontenc}
11382 \else
11383 \usepackage{fontspec}
11384 \fi
11385
11386 \begin{document}
11387 \microtypesetup{expansion=false}
11388
11389 %% load your font here:
11390
11391 \ShowCharacterInheritance
11392
11393 \newpage
11394 \ShowProtrusion
11395
11396 \newpage
11397 %% show single glyphs
11398 %\ShowDummyLine
11399 %\ShowProtrusionLineGlyph{A}
11400 %\ShowProtrusionLineIndex{27}
11401
11402 %% loop through all glyphs of the font;
11403 %% protrusion values are shown in 1000th of 1em
11404 \ShowProtrusionDefined
11405
11406 %\ShowProtrusionMissing
11407
11408 %\ShowProtrusionAll
11409
11410 \newpage %% -----
11411 This is the current font stretched by 5%, normal, and shrunk by 5%:
11412
11413 \newlength{\MTln}
11414 \newcommand*{\teststring}
11415 {ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyz}
11416 \settowidth{\MTln}{\teststring}
11417 \microtypesetup{expansion=true}
11418
11419 \bigskip\noindent\parbox{1.05\MTln}{\teststring\linebreak\\\teststring}\par
11420 \bigskip\noindent\parbox{0.95\MTln}{\teststring}
11421 \end{document}
11422 (/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/schlicht/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}
```

11423 *(*logo)*

Here's how the logo on the title page was created.²³ 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.²⁴ 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`.

11424 `\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.

11425 `\input bbox.sty`

`\tempdim` Allocate some `dimen` registers.

11426 `\newdimen\tempdim`

`\fboxrulei` Frame width of the box as \TeX sees it.

11427 `\newdimen\fboxrulei`

11428 `\fboxrulei=0.1pt`

`\fboxruleii` Frame width of the bounding box.

11429 `\newdimen\fboxruleii`

11430 `\fboxruleii=0.1pt`

`\kernboxheight` Height of the box indicating the kern.

11431 `\newdimen\kernboxheight`

11432 `\kernboxheight=5pt`

`\scaletoem` An auxiliary macro. Return a dimension relative to the `em`-width of the font. Requires `e-TeX`.

11433 `\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).

11434 `\fontinstcc`

11435 `\def\showlogo#1{%`

Some fonts do not specify the `\fontdimen6` (width of an `em`) in the `afm` file. In this case, use the font size, which is correct in most cases.

11436 `\ifdim\fontdimen6\font = 0pt`

11437 `\typeout{***-Warning:-no-fontdimen-6-specified-***^^J%`

11438 `***-setting-it-to-\pdffontsize\font \ifnum\pdfversion < 130 pt\fi-***}`

11439 `\fontdimen6\font=\pdffontsize\font \ifnum\pdfversion < 130 pt\fi\relax`

11440 `\fi`

11441 `\installfonts`

11442 `\input_metrics{}{\logofont,\metrics\printbbs{#1}\relax}`

23 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.

24 Message ID: 42aa3687\$0\$24366\$9b4e6d93@newsread2.arcor-online.net

```

11443 \endinstallfonts
11444 }
11445 \normalcc
      Layers.
11446 \makeatletter
11447 \def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
11448 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
11449 \ifx\mt@order \undefined\let\mt@order \@empty\fi
11450 \xdef\mt@order{\mt@order[(Logo)]}
11451 \let\mtl@resources\@empty
11452 \def\mtl@register#1{%
11453   \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
11454   \expandafter\xdef\csname mtl@#1\endcsname{\the\pdflastobj\space 0 R }
11455   \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
11456   \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
11457   \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
11458 \mtl@register{canvas}
11459 \mtl@register{characters}
11460 \mtl@register{bounding-boxes}
11461 \mtl@register{TeX-boxes}
11462 \xdef\mt@order{\mt@order]}
11463 \global\let\mtl@objects\mt@objects
11464 \def\togglelayer#1#2{%
11465   \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
11466   user{/Subtype/Link
11467     /BS << /Type/Border/W 0 >> /H/0
11468     /A << /S/SetOCGState
11469     /State[/Toggle \csname mtl@#1\endcsname] >>
11470   }#2\pdfendlink
11471 }

```

\printbbs Preparation.

```

11472 \setcommand\printbbs#1{%
11473   \setbox0\hbox{#1}%
11474   \leavevmode
11475   \kern-\fboxrulei
      The canvas in the natural width of the text minus protrusion, in color bgcolor.
11476   \mtl@layer{canvas}{%
11477     \getboundarychars#1\relax
11478     \tempdim=\dimexpr\wd0 - (\scalettoem{\lpcode\font\firstchar}+
11479       \scalettoem{\rptide\font\lastchar})\relax
11480     \kern\dimexpr\scalettoem{\lpcode\font\firstchar}\relax
11481     \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
11482       \hrule width \tempdim
11483         height \dimexpr\dp0+\ht0+0.15em\relax}%
11484     \kern-\tempdim

```

The baseline, in color blcolor.

```

11485     \vbox{\color{blcolor}%
11486       \hrule width \tempdim
11487         height \fboxrulei}%
11488     }%
11489     \kern-\dimexpr\wd0 -\scalettoem{\rptide\font\lastchar}\relax

```

The string.

```

11490   \printbbs #1\relax\relax
11491 }

```

\getboundarychars Get first

```

11492 \def\getboundarychars#1#2\relax{%
11493   \def\firstchar{~#1}%
11494   \getlastchar#1#2\relax
11495 }

```

\getlastchar ... and last character.

```

11496 \def\getlastchar#1#2{%

```

```

11497 \ifx\relax#2\relax
11498 \def\lastchar{`#1}%
11499 \else
11500 \expandafter\getlastchar
11501 \fi #2%
11502 }

```

`\printbss` Loop over all characters of the string.

```

11503 \def\printbss#1#2#3\relax{%
11504 \ifx\relax#1\relax
11505 \else
11506 \ifx\relax#2\relax
11507 \printbb{#1}{}%
11508 \else
11509 \printbb{#1}{#2}%
11510 \fi
11511 \expandafter\printbss
11512 \fi #2#3\relax
11513 }

```

`\printbb` Record the kern between the current and the following character, then print the character. `\kerning` is a fontinst command.

```

11514 \setcommand\printbb#1#2{%
11515 \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
11516 \showboxes{#1}%

```

This could be another application.

```

11517 % \quad
11518 % w: \the\scaletoe{\width{#1}},
11519 % bb: \the\scaletoe{\bbleft{#1}}/%
11520 % \the\scaletoe{\bbright{#1}},
11521 % \the\scaletoe{\number\numexpr\width{#1}-\bbright{#1}\relax}
11522 % h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
11523 }

```

`\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' ≠ ',').

```

11524 \setcommand\showboxes#1{%
11525 \leavevmode
11526 \color{texcolor}%

```

We have to record the width of the glyph.

```

11527 \setbox0\hbox{\color{textcolor}#1}%
11528 \global\tempdim=\wd0\relax
11529 \kern-\fboxrulei

```

1. *The TeX box*: Print a frame in color `texcolor`. This frame shows the glyph as TeX sees it.

```

11530 \mtl@layer{TeX-boxes}{%
11531 \hbox{%
11532 \lower\dimexpr \dp0 + \fboxrulei\relax
11533 \hbox{%
11534 \vbox{%
11535 \hrule height\fboxrulei
11536 \hbox{%
11537 \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
11538 \phantom{\unhcopy0}%
11539 \vrule width\fboxrulei
11540 }%
11541 \hrule height\fboxrulei}}}%
11542 }%

```

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.

```

11543 \kern-\wd0
11544 \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.

```

11545 \kern\dimexpr\scaletoe{\bbleft{#1}}-\tempdim-\fboxrulei\relax

```

3. *The bounding box*: will be printed in color `bbcolor`.

```

11546 \mtl@layer{bounding-boxes}{%
11547   {\color{bbcolor}%
11548   \hbox{%
11549     \lower\dimexpr-\scaletoe{\bbbottom{#1}}+\fboxruleii\relax
11550     \hbox{%
11551       \vbox{%
11552         \hrule height\fboxruleii
11553         \hbox to \dimexpr\scaletoe{\numexpr
11554           \bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{%
11555           \vrule height \dimexpr\scaletoe{\numexpr
11556             \bbtop{#1}-\bbbottom{#1}\relax}%
11557             width\fboxruleii
11558             \hfill
11559             \vrule width\fboxruleii}%
11560           \hrule height\fboxruleii}}}%
11561     }%
11562     \kern-\dimexpr\fboxruleii+\fboxrulei\relax
11563   }%

```

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.

```

11564 \kern\scaletoe{\numexpr\width{#1}-\bbright{#1}\relax}%
11565 \mtl@layer{TeX-boxes}{%
11566   {\ifnum\thekern<0
11567     \color{kerncolor}%
11568     \kern\scaletoe{\thekern}%
11569     \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoe{\thekern}\relax
11570       height \kernboxheight}%
11571     \kern\scaletoe{\thekern}%
11572   }else
11573     \color{texcolor}%
11574     \ifnum\thekern=0 \else
11575       \lower\kernboxheight
11576       \hbox{%
11577         \vbox{%
11578           \hrule height\fboxruleii
11579           \hbox{%
11580             \vrule height \kernboxheight width\fboxruleii
11581             \kern\dimexpr\scaletoe{\thekern}-2\fboxrulei\relax
11582             \vrule width\fboxruleii
11583           }%
11584           \hrule height\fboxruleii}}}%
11585     \fi
11586     \fi
11587   }%
11588 }%
11589 % \kern-\fboxrulei
11590 }

```

\printlogo

```

11591 \newbox\logobox
11592 \def\printlogo{%
11593   \setbox\logobox=\hbox{\vbox{%
11594     \MakePercentComment

```

This is the Kepler MM font used in the logo.

```

11595   \def\logofont{pkpri9e10}
11596   \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
11597   \font\thelogofont=\logofont\space at 82pt

```

This would load the italic Palatino font instead.

```

11598 %\def\logofont{pplri}
11599 %\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
11600 %\edef\logofont{\logofont8r}
11601 %\font\thelogofont=\logofont\space at 78pt

```

Load the font.

```
11602 \thelogo font
```

Protrusion values (overdone for didactic reasons).

```
11603 \lcode\font`M=96
```

```
11604 \rcode\font`e=46
```

Now we can generate the logo.

```
11605 \pdfliteral direct{/SXS gs}%
```

```
11606 \showlogo{Microtype}%
```

```
11607 % \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
```

```
11608 % \kern5pt\[\[3\baselineskip]
```

```
11609 % \long\def\@makefnmark##1{%
```

```
11610 % \leftskip 0pt
```

```
11611 % \parindent 0pt
```

```
11612 % \everypar{\parindent 0pt}%
```

```
11613 % \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
```

```
11614 % \footnotetext[1]{This graphic displays on a
```

```
11615 % \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
```

```
11616 % their \togglelayer{bounding-boxes}{bounding boxes}
```

```
11617 % and \togglelayer{TeX-boxes}{\TeX\ boxes}.
```

```
11618 }}%
```

```
11619 \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
```

```
11620 \immediate\pdfobj<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>%
```

```
11621 \immediate\pdfxform
```

```
11622 attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
```

```
11623 resources {/Properties <<\mtl@resources>>
```

```
11624 /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
```

```
11625 \logobox
```

```
11626 % \vskip-2.5\baselineskip
```

```
11627 % \leavevmode
```

```
11628 % \togglelayer{characters}{%
```

```
11629 % \pdfrefxform\pdflastxform
```

```
11630 % }%
```

```
11631 \pdfannot\logodimens{%
```

```
11632 /Subtype/Widget /FT/Btn /T(Logo)
```

```
11633 %/F 4 % why did I say this?
```

```
11634 /AP << /N \the\pdflastxform\space 0 R >>
```

```
11635 /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
```

```
11636 /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
```

```
11637 /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
```

```
11638 /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
```

```
11639 >> }%
```

```
11640 \vspace{3\baselineskip}
```

```
11641 }
```

```
11642 \IfFileExists{pkpmmri8a10.afm}\relax{\def\printlogo{\MT@warning{File pkpmmri8a10.afm not found.}}
```

```
11643 \MessageBreak Cannot create logo}}}
```

Our font.

```
11644 \pdfmapline{+pkpmmri8r10 Kep1MM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
```

Define colours (thered and thegreen are copied from microtype.dtx).

```
11645 \def\mtdefinecolors{
```

```
11646 \definecolor{thered}{rgb}{0.65,0.04,0.07}
```

```
11647 \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
```

```
11648 \colorlet{texcolor}{thegreen!50} % TeX boxes
```

```
11649 \colorlet{kerncolor}{texcolor} % negative kerns
```

```
11650 \colorlet{bbcolor}{thered!50} % bounding box
```

```
11651 \colorlet{bgcolor}{black!8} % canvas
```

```
11652 \colorlet{blcolor}{black!50} % baseline
```

```
11653 \colorlet{textcolor}{black!40} % text
```

```
11654 }
```

Use with microtype.dtx

```
11655 \ifx\documentclass\@twoclasseserror
```

```
11656 \usepackage[xcdraw]{xcolor}
```

```
11657 \mtdefinecolors
```

```
11658 \else
```

A.2 Document

Now we can start the document.

```
11659 \documentclass[10pt,a4paper]{ltxdoc}
11660 \providecommand\MakePercentComment{\relax}
11661 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
    Re-use the preamble from microtype.dtx.
11662 \usepackage{microtype-doc}
11663 \usepackage{attachfile}
11664 \makeatletter
11665 \pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
11666 \makeatother
11667 \begin{document}
    You are currently reading this.
11668 \DocInput{microtype-logo.dtx}
11669 \newpage
11670 And here it is:\vspace{6\baselineskip}
11671 \begin{center}
11672   \printlogo
11673 \end{center}
11674 \expandafter\enddocument
11675 \fi
    That's it.
11676 </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}
```

```
11677 \ifx\lssample\undefined
11678 <*\lssample>
```

Upon popular request, here's how I've created the letterspacing illustration.²⁵

B.1 Macros

Rule width and image height and depth.

```
11679 \makeatletter
11680 \newdimen\lsamount
11681 \newdimen\lsrule
11682 \lsrule=0.2pt
11683 \def\lsheight{8pt}
11684 \def\lsdepth{12pt}
```

²⁵ 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).

```

11685 \def\lsfont{\fontfamily{paca}\selectfont}
      Loop over all letters in <#2>, letterspacing them by <#1>.
11686 \def\dols#1#2{\lssamount=#1\relax \dols#2\enddols}
11687 \def\dolss#1#2\enddols{%
11688   \ifx\empty#2\empty\divide\lssamount 2\fi
11689   \ls{#1}%
11690   \ifx\empty#2\empty\else \dolss#2\enddols \fi
11691 }

      One tikz picture for each letter.
11692 \def\ls#1{%
11693   \begin{tikzpicture}[remember picture,line width=\lsrule]
11694     \tikzstyle{every node}=[inner sep=0pt]

      The bounding box.
11695     \mts@layer{stuff}{%
11696       \node[draw=thegrey,
11697         fill=theshade,
11698         outer sep=\lsrule,
11699         anchor=base,
11700         font=\lsfont]{\phantom{#1}};
11701     }

      The letter.
11702     \node[anchor=base,font=\lsfont](#1){#1};

      Two auxiliary coordinates.
11703     \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
11704     \path (#1.base east) ++(-.5\lsrule,-\lsdepth) coordinate (#1R);
11705     \mts@layer{stuff}{%

      Now draw the normal character width,
11706       \draw[color=thered!75,
11707         fill=thered!30,
11708         outer sep=\lsrule]
11709         (#1L) rectangle (#1R);
11710       \ifdim\lssamount>0pt
11711         \path (#1.base east) ++(+.5\lssamount,-6pt) coordinate (#1_1s);
11712         \path (#1R) ++(\lssamount+\lsrule,\lsdepth) coordinate (#1E);

      and the letter space.
11713         \draw[color=thered,
11714           fill=thered!50,
11715           outer sep=\lsrule]
11716           (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
11717         \fi
11718       }
11719     \end{tikzpicture}%
11720     \ignorespaces
11721 }

      Draw the interword space.
11722 \def\lssp#1#2#3#4{%
11723   \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
11724     \mts@layer{stuff}{%
11725       \tikzstyle{every draw}=[anchor=bottom]
11726       \coordinate(#1space) at (#2/2,\lsdepth/2);
11727       \coordinate(#1stretch) at (#2+#3/2,+0pt);
11728       \coordinate(#1shrink) at (#2-#4/2,+0pt);
11729       \draw[color=thegreen,fill=thegreen!50,use as bounding box]
11730         (0,0) rectangle ++(#2,\lsdepth);
11731       \draw[color=thegreen,fill=thegreen!30]
11732         (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
11733       \draw[color=thegreen,fill=thegreen!50]
11734         (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
11735       \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]

```



```

11736      (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
11737      \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
11738      (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
11739      }%
11740 \end{tikzpicture}%
11741 \ignorespaces
11742 }

Layers.
11743 \def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
11744 \def\mts@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
11745 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
11746 \ifx\mt@order \undefined\let\mt@order \@empty\fi
11747 \xdef\mt@order{\mt@order[(Sheep)]}
11748 \let\mts@resources\@empty
11749 \def\mts@register#1{%
11750 \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
11751 \expandafter\xdef\csname mts@#1\endcsname{\the\pdfastobj\space 0 R }
11752 \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
11753 \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
11754 \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
11755 \mts@register{stuff}
11756 \mts@register{tracking}
11757 \mts@register{ispace}
11758 \mts@register{ospace}
11759 \mts@register{istretch}
11760 \mts@register{ishrink}
11761 \mts@register{ostretch}
11762 \mts@register{oshrink}
11763 \mts@register{okern}
11764 \mts@register{ligature}
11765 \mts@register{_compatibility}
11766 \xdef\mt@order{\mt@order]}

Anchor point for the arrow in the code.
11767 \newcommand\anchorarrow[1]{%
11768 \tikz[remember picture,overlay]\node(#1_c){};}

Add an arrow from code to image.
11769 \newcommand\add@arrow[5][left]{%
11770 \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
11771 \mts@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}%
11772 }

Toggle layer.
11773 \def\toggle@layer#1#2#3{%
11774 \pdfstartlink
11775 user{/Subtype/Link
11776 /BS << /Type/Border/W 0 >> /H/0
11777 % /BS << /Type/Border/W 1 /S/D /D[4 1] >>
11778 % /C[0.7 0.7 0.7] /H/0
11779 /Contents(Click to Toggle!)
11780 /A << /S/SetOCGState
11781 /State[/Toggle \csname mts@#1\endcsname] >> }%
11782 \rlap{#2}%
11783 {\fboxsep=0pt \fboxrule=0pt
11784 \mts@layer{stuff}{%
11785 \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
11786 \mts@layer{#1}{%
11787 \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}}%
11788 }%
11789 \pdfendlink
11790 }
11791 \newcommand\showarrow[2][ ]{%
11792 \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
11793 \toggle@layer{\@tempa}{\itshape #2}}

```

The environment for our illustration.

```

11794 \def\ls@sample#1{%
11795   \parskip 4pt \parindent 0pt
11796   \par
11797   \vskip4pt
11798   {\leftskip 15pt
11799   \mt@pseudo@margin{\color{theblue}Click on the image to show the kerns
11800     and spacings involved. Click on emphasised words in the text below
11801     to reveal the relation of image and code.\strut}
11802   \mt@layer{_compatibility}{%
11803     \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
11804       \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
11805     \mt@pseudo@margin{\color{thered}%
11806       If you had a \acronym{PDF} viewer that understands
11807       \acronym{PDF}\,\{\smaller1.5}, you could hide the arrows selectively.}}
11808     \vskip-\mt@unvdimen}%
11809   \vskip-4pt
11810   \setlength\fbxsep{4pt}%
11811   \leavevmode
11812   \pdfstartlink
11813     user{/Subtype/Link
11814       /BS << /Type/Border/W 0 >> /H/0
11815       /A << /S/SetOCGState
11816         /State[/Toggle \mts@stuff] >> }%
11817     \fcolorbox{theframe}{theshade}%
11818     {\fontsize{34}{38}\selectfont #1}%
11819   \pdfendlink
11820   \par\medskip
11821   }%
11822   \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
11823 }

```

Now define the illustration to be used in the document.

```

11824 \def\lssample{%
11825   \ls@sample{%
11826     \dols{Opt}{Stop}
11827     \lssp{o}{0.45em}{0.25em}{0.15em}
11828     \dols{0.16em}{\stearing}\hskip-\dimexpr 0.08em+\lslrule\relax
11829     \lssp{i}{13.82pt}{4.65pt}{2.08pt}
11830     \dols{0.16em}{sheep}
11831     \dols{Opt}{!}
11832   }%

```

Don't forget to add the arrows.

```

11833   \vspace{-\baselineskip}
11834   \add@arrow{red} {tracking}{\lsamount_c.east}{a_ls}
11835   \add@arrow{red} {okern} {okernend_c.east}{p_ls}
11836   \add@arrow{green} {ospace} {ospace_c.east} {ospace}
11837   \add@arrow{green} {ispace} {ispace_c.center} {ispace}
11838   \add@arrow{green!75} {istretch} {istretch_c.east} {istretch.north}
11839   \add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
11840   \add@arrow{green!75} {ostretch} {ostretch_c.east} {ostretch.north}
11841   \add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
11842   \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
11843 }
11844 \fi

```

This is for use with microtype.dtx

```

11845 \ifx\documentclass\@twoclasseserror
11846   \usepackage{tikz}
11847 \else

```

B.2 Document

```

11848 \documentclass[10pt,a4paper]{ltxdoc}
11849 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

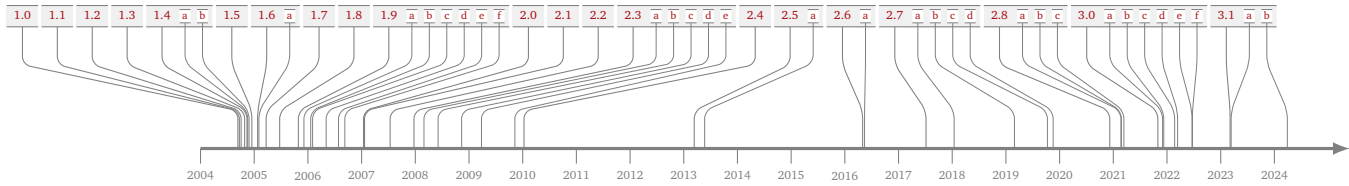
```

```

Re-use the preamble from microtype.dtx.
11850 \usepackage{microtype-doc}
11851 \usepackage{attachfile}
11852 \usepackage{tikz}
11853 \makeatletter
11854 \pdfcatalog{/OCProperties << /OCGs [\mt@objects]
11855                               /D << /Order [\mt@order] /BaseState/OFF >> >> }
11856 \makeatother
11857 \begin{document}
You are currently reading this.
11858 \DocInput{microtype-1ssample.dtx}
Now show what we are able to do.
11859 \noindent
11860 Since a picture is worth a thousand words, probably even more if, in our
11861 case, it depicts a couple of letterspaced words, let's bring one to sum up
11862 these somewhat confusing options. Suppose you had the following settings
11863 (which I would in no way recommend; they are only for illustrative purposes):
11864 \begin{verbatim}
11865 \SetTracking
11866 [ no ligatures = {"\anchorarrow{nolig}"f},
11867   spacing      = {60"\anchorarrow{ispace}"0*,"%
11868                 "-1"\anchorarrow{istretch}"00*," \anchorarrow{ishrink}"},
11869   outer spacing = {4"\anchorarrow{ospace}"50,"%
11870                 "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
11871   outer kerning = {"\anchorarrow{okernbegin}"*,"%
11872                 \anchorarrow{okernend}"* } ]
11873 { encoding = * }
11874 { 1"\anchorarrow{lsamount}"60 }
11875 \end{verbatim}
11876 and then write:
11877 \begin{verbatim}
11878 Stop \textls{stealing sheep}!
11879 \end{verbatim}
11880 this is the (typographically dubious) outcome:
11881
11882 \lssample
11883
11884 \noindent
11885 While the word 'Stop' is not letterspaced, the space between the letters in
11886 the other two words is expanded by the \showarrow[tracking]{tracking-amount}{red}
11887 of 160/1000\,em\,=\allowbreak\,0.16\,em.
11888 The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
11889 increased by 60\%, while its \showarrow[istretch]{stretch}{green} amount is
11890 decreased by 10\% and the \showarrow[ishrink]{shrink}{green} amount is left
11891 untouched.
11892 The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
11893 piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
11894 \showarrow[oshrink]{shrink}{green} by 0.15\,em.
11895 Note that there is no outer space after the text, since the exclamation mark
11896 immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
11897 of half the letterspace amount (0.08\,em) is added.
11898 Furthermore, one \showarrow{ligature}{grey} wasn't broken up, because we
11899 neglected to specify the '~|s|' in the |no ligatures| key.
11900
11901 \expandafter\enddocument
11902 \fi
11903 </lssample>

```

C Change history



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 <i>Harald Harders</i>)	77	<code>\MT@get@listname@</code> : don't check for empty attributes list	79
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i>)	141	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i>)	20
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by <i>Harald Harders</i>)	149	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i>)	85
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance	150	<code>\MT@pdftex@no</code> : fix: version check (reported by <i>Harald Harders</i>)	15
<code>\MT@declare@sets</code> : remove spaces around set name	111	<code>\MT@permute</code> : don't use sets for empty encoding	113
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded	97	<code>\MT@setup@expansion</code> : issue an error instead of a warning, when pdfTeX version is too old for autoexpand	130
<code>\MT@get@basefamily</code> : only remove suffixes 'x' or 'j'	77	<code>\MT@split@codes</code> : fix: allow zero and negative values	44
	79	<code>\MT@use@set</code> : remove spaces around set name	101

2004/10/03 **Version 1.2**

Font aliases: declare cmor as an alias of cmr	138	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to \empty	81
Font sets: new: allmath and basicmath	137	<code>\MT@get@listname@</code> : alternatively check for alias font name	79
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding	180	<code>\MT@get@size</code> : additional magic to catch some errors	99
add settings for Computer Modern Roman math symbols	184	<code>\MT@get@size@</code> : hijack \set@fontsize instead of \setfontsize	100
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement	40	<code>\MT@loop</code> : fix: new macro, used instead of \loop	24
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals)	79	<code>\MT@maybe@do</code> : also check for alias font name	40
<code>\MT@get@highlevel</code> : check whether defaults have changed	98	<code>\MT@permute@@@@</code> : more sanity checks for \SetProtrusion and \SetExpansion	115
		<code>\MT@setupfont</code> : also search for alias font file	37
		fix: call \@enc@update if necessary	37

2004/10/27 **Version 1.3**

General: fix: specifying load option does no longer require to give a name, too	108	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german)	5
Font aliases: declare aer, zer and hfor as aliases of cmr	138	<code>\MT@load@list</code> : check whether list exists	77

2004/11/12 **Version 1.4**

General: check for pdfcpot	30	the hook for <code>\MT@setupfont</code>	91
don't use scratch registers in global definitions	81	use one instead of five counters	26
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as		Protrusion: tweak quote characters for cmr variants	

- (OT1, T1, lmr) 155
- `\microtypesetup`: fix: set the correct levels, and remember them; warning when enabling an option 124
- 2004/11/17 **Version 1.4a**
- General: new option: `final` 120
- `\MT@cfg@catcodes`: fix: reset some more catcodes 124
- 2004/11/26 **Version 1.4b**
- General: fix: set catcodes before reading global configuration file (reported by *Christoph Bier*) 122
- optimisation: use less `\expandafers` and `\csnames` 19
- Protrusion: harmonise dashes in upshape and italic (`cmr`, `pad`, `pp1`) 149
- slanted like italics 160
- `\MT@checklist@family`: fix: don't try alias family name if encoding failed 41
- `\MT@get@basefamily`: fix: failed for font names of the form `abczz` (reported by *Georg Verwey*) 79
- `\MT@get@slot`: don't define `\MT@char` globally (save stack problem) 81
- `\MT@ifdimen`: don't set `\MT@count` globally (save stack problem) 21
- `\MT@setup@PDF`: new message if `\pdfoutput` is changed 128
- `\MT@use@set`: don't use undeclared font sets 101
- 2004/12/15 **Version 1.5**
- General: defaults: step: 4 (suggested by *Hàn Thế Thành*) 120
- new option: selected, by default false (suggested by *Hàn Thế Thành*) 118
- Documentation: add 'Short history' U32
- Inheritance: remove `\ss` from T1 list, add `\DJ` 142
- Protrusion: settings for Bitstream Charter 150
- `\DeclareMicrotypeAlias`: remove spaces around arguments 103
- `\MT@cfg@catcodes`: reset catcode of '=' (compatibility with Turkish `babel`) 78
- `\MT@fix@catcode`: reset catcode of '^' (compatibility with `chemsym`) 5
- `\MT@get@highlevel`: don't test defaults if called after begin document 98
- `\MT@scale@factor`: warning for factors outside limits 46
- `\MT@scale@to@em`: don't use `\lcode` and `\rcode` for the calculation 45
- `\MT@set@ex@codes`: allow non-selected font expansion 59
- `\MT@set@pr@codes`: adjust protrusion factors before setting the inheriting characters 42
- `\MT@setup@expansion`: defaults: calculate step as $\min(\text{stretch}, \text{shrink})/5$ 129
- defaults: turn off expansion for DVI output 128
- disable automatic expansion for DVI output 130
- 2005/01/24 **Version 1.6**
- General: defaults: turn off expansion for old pdfTeX versions 122
- load a font if none is selected 37
- new option: factor, by default 1000 120
- restructure dtx file 137
- test whether `\pickup@font` has changed 93
- test whether numeric options receive a number 120
- use e-TeX's `\ifcname` and `\ifdefined` if defined 20
- Protrusion: add italic uppercase Greek letters 160
- improve settings for numbers (pointed out by *Peter Muthesius*) 151
- tune CMR math letters (OML encoding) 185
- `\MT@get@charwd`: use e-TeX's `\fontcharwd`, if available 45
- `\MT@get@inh@list`: correct message if selected is false 81
- `\MT@set@ex@codes`: introduce factor option 59
- `\MT@set@pr@codes`: introduce factor option 42
- `\MT@setup@expansion`: disable automatic expansion for old pdfTeX versions 130
- `\MT@use@set`: retain current set if new set is undeclared 101
- `\MT@vinfo`: new macro instead of `\ifMT@verbose` 6
- 2005/02/02 **Version 1.6a**
- Documentation: add table of fonts with tailored protrusion settings U21
- `\MT@get@slot`: completely redone, hopefully more robust (compatible with `frenchpro`; problem reported by *Bernard Gaulle*) 81
- `\MT@pdfTeX@no`: new macro 14
- `\MT@reset@ef@codes`: only reset `\efcodes` for older pdfTeX versions 60

2005/03/23 **Version 1.7**

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i>)	98	<code>\MT@get@slot</code> : remove backslash hack	81
disallow automatic expansion if pdfTeX too old	110	test for <code>\chardefed</code> commands	82
fix: remove space after <code>autoexpand</code>	110	test whether <code>\(encoding)\(…)</code> is defined	82
new value for <code>verbose</code> option: errors	120	<code>\MT@if@list@exists</code> : don't define <code>\MT@pr@c@name</code> etc. globally, here and elsewhere	80
shorter command names	26	<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i>)	21
warning when running in draft mode	127	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	26
Documentation: add hint about compatibility	U27	<code>\MT@is@composite</code> : new macro: construct command for composite character; no uncontrolled expansion	88
remove table of match order (now table 1 on page 79)	U11	<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if available	26
Protrusion: fix: remove <code>\</code> from OT1, add <code>\textbackslash</code> to T1 encoding	152	<code>\MT@set@ex@codes</code> : two versions of this macro	59
<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i>)	103	<code>\MT@split@name</code> : don't define <code>\MT@encoding</code> &c. globally	40
<code>\Microtype@Hook</code> : new command for font package authors	123	<code>\MT@test@ast</code> : make it simpler	98
<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	124	<code>\MT@try@order</code> : always check for size, too (suggested by <i>Andreas Böhmann</i>)	79
<code>\MT@begin@catcodes</code> : also use inside configuration commands	78	fix: also check for <code>//(series)/(shape)//</code> (reported by <i>Andreas Böhmann</i>)	79
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'</code> (compatibility with french* packages)	78	<code>\MT@warn@code@too@large</code> : new macro: type out maximum protrusion factor	47
<code>\MT@DeclareMicrotypeAlias</code> : may also be used inside configuration files	103	<code>\MT@warn@err</code> : new macro: for <code>verbose=errors</code>	6
<code>\MT@get@listname@</code> : use <code>\@tfor</code> (<i>Andreas Böhmann's</i> idea)	79	<code>\showhyphens</code> : modify <code>\showhyphens</code>	131

2005/06/23 **Version 1.8**

General: <code>\SetProtrusion</code> : new key: <code>unit</code> if font substitution has occurred, set up the substitute font, not the selected one	109	<code>\MT@find@file</code> : no longer wrap names in commands	77
new option: <code>config</code> to load a different main configuration file	122	<code>\MT@fix@fontdimen@six</code> : new macro: test whether <code>\fontdimen 6</code> is defined	39
new option: <code>unit</code> , by default character	121	<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters	45
Documentation: add example for <code>factor</code> option	U12	<code>\MT@get@listname@</code> : made recursive	79
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i>)	U14	<code>\MT@get@slot</code> : fix: expand active characters	81
add hint about error messages	U28	test whether <code>\(encoding)\(…)</code> is defined made more robust	82
Font aliases: declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	139	<code>\MT@get@unit</code> : new macro: get unit for codes	48
Font sets: add U encoding to <code>allmath</code>	137	<code>\MT@in@rlist</code> : made recursive	24
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as <code>\DH</code>)	142	<code>\MT@is@active</code> : new macro: translate inputenc-defined characters	85
Protrusion: add LY1 characters for Times	158	<code>\MT@is@letter</code> : warning for non-ASCII characters	84
settings for AMS math fonts	188	<code>\MT@ledmac@setup</code> : character protrusion with <code>ledmac</code>	28
verified settings for slanted Computer Modern Roman	170	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code>	23
<code>\add@accent</code> : fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by <i>Stephan Hennig</i>)	93	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code>	23
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font	103	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code>	6
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set	102	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30	14
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining ASCII characters	78	<code>\MT@permute@#####</code> : add ranges to the beginning of the lists	115
<code>\MT@check@rlist</code> : made recursive	116	<code>\MT@scale</code> : fix: remove spaces in ϵ -TeX variant (reported by <i>Mark Rossi</i>)	26
<code>\MT@curr@list@name</code> : new macro: current list type and name	89	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>hyperref</code> is loaded	30
<code>\MT@declare@sets</code> : warning when redefining a set	97	restore <code>csquotes's</code> active characters	29
<code>\MT@define@set@key@</code> : use comma lists instead of token lists	97	restore percent character if Spanish <code>babel</code> is loaded	29
		<code>\MT@split@codes</code> : get character width once only	44
		<code>\MT@use@set</code> : fix: remove braces in first line	101
		<code>\MT@xadd</code> : simplified	23

2005/10/28 **Version 1.9**

General: <code>\DeclareMicrotypeSet</code> : new key: font	100	settings for T5 encoded Computer Modern Roman	149
<code>\SetProtrusion</code> : value ‘relative’ renamed to ‘character’ for key unit	109	<code>\DisableLigatures</code> : new command: disable ligatures (requires pdf \TeX 1.30)	104
allow context-specific font setup	91	<code>\microtypecontext</code> : new command: change setup context in the document	95
compatibility with \TeX Live hack (reported by <i>Herbert Voß</i>)	13	<code>\MT@checklist@family</code> : fix: add two missing <code>\expandafters</code>	41
disable microtype setup inside <code>hyperref</code> ’s <code>\pdfstringdef</code> (reported by <i>Hàn Thê Thành</i>)	31	<code>\MT@detokenize@c</code> : fix the ϵ - \TeX version	20
fix: use true as the default value	117	<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code>	19
option unit: rename value relative to character	121	<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists	48
Documentation: add remark about Type 1 fonts required for automatic font expansion	U7	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code>	85
Font aliases: declare <code>qp1</code> and <code>qtm</code> (qfonts, \TeX Gyre) as aliases of <code>pp1</code> resp. <code>ptm</code>	139	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code>	84
Font sets: add OT4 encoding to text sets	137	<code>\MT@maybe@do</code> : redone	40
add T5 encoding to text sets	137	<code>\MT@rem@from@clist</code> : new macro: remove an item from a comma list	24
Inheritance: add list for OT4	143	<code>\MT@scale@factor</code> : generalised	46
add list for T5 (requested by <i>Hàn Thê Thành</i>)	144	<code>\MT@setup@expansion</code> : disable expansion if both step and shrink are zero	130
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	153	warning if user requested zero step	129
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	149	<code>\MT@toks</code> : use instead of <code>\toks@</code>	16
		<code>\SetProtrusion</code> : (et al.) new key: font	104

2005/12/05 **Version 1.9a**

General: ‘ <i>file name</i> ’/‘ <i>line number</i> ’ as default list name	108	diately (requested by <i>Georg Verweyen</i>)	97
new option: <code>deferssetup</code> , by default true	119	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed	98
remove superfluous test whether <code>\pickup@font</code> has changed	93	<code>\MT@ifdefined@c@T</code> : new macros: true case only	20
Documentation: add explanation for error message in DVI mode	U29	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available	21
add explanation for error message with non-Type 1 fonts	U29	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available	22
Font aliases: declare <code>mbch</code> (mathdesign) as an alias of <code>bch</code>	140	<code>\MT@in@clist</code> : fix	24
Protrusion: fix: remove ‘_’ from OT1 encoding	154	<code>\MT@info@missing@char</code> : info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC)	46
settings for T5 encoded Charter	149	<code>\MT@is@feature</code> : new macro: check for pdf \TeX feature	27
<code>\microtypesetup</code> : inside the preamble, accepts all package options	124	<code>\MT@map@clist@n</code> : following \LaTeX 3	23
<code>\MT@check@font@cx</code> : optimise context-sensitive setup	94	<code>\MT@permute@ooo</code> : don’t define permutations for unused encodings	114
<code>\MT@define@set@key@</code> : don’t expand variables immediately (requested by <i>Georg Verweyen</i>)	97	<code>\MT@rem@from@clist</code> : fix	24
		<code>\MT@setup@</code> : defer setup until the end of the preamble	27

2006/01/20 **Version 1.9b**

General: compatibility with <code>listings</code> : sanitise more catcodes (reported by <i>Holger Uhr</i>)	31	add samples of micro-typographic features	U3
compatibility with the <code>extendedchar</code> option of the <code>listings</code> package	31	<code>\MT@features</code> : use throughout the package to adjust to beta-ness	27
Documentation: activate expansion in the distributed PDF	U1	<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available	21
		<code>\MT@warn@code@too@large</code> : fix calculation with present factor	47

2006/02/02 **Version 1.9c**

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verweyen</i>)	U22	<code>\MT@define@code@key@font</code> : fix: context was ignored	107
Protrusion: settings for URW Garamond	150	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@temp@size</code> in <code>\csname</code> (bug introduced in v1.9b)	107

2006/05/05 **Version 1.9d**

Font sets: md* instead of m series in basic sets	137	\MT@get@font@dimen: warning for zero fontdimen . .	46
add QX encoding to text sets	137	\MT@get@opt: optimise: don't reset when preset op-	48
Inheritance: add list for QX encoding (contributed by		tion is set	48
<i>Maciej Eder</i>)	144	set list name before presetting	48
Protrusion: settings for QX encoding (contributed by		\MT@is@active: support for Unicode (inputenc/utf8) 85	
<i>Maciej Eder</i>)	157	\MT@setupfont@hook: restore % and \# when tex4ht	
settings for Euro symbols (Adobe, ITC, marvosym)	196	is loaded (reported by <i>Peter Dyballa</i>)	30
tweak AMS settings	188	\SetProtrusion: (et al.) optimise: unify keys for	
\DeclareCharacterInheritance: fix: empty context	111	mandatory argument	104
\MT@detokenize@n: new macro: use \detokenize if		(et al.) split keys of optional and mandatory argu-	
available	20	ment	104
\MT@get@ex@opt: fix: evaluate preset	61		

2006/07/28 **Version 1.9e**

General: fix: default value for activate: true . . .	118	settings for Euler Roman font	191
Documentation: add hint about unknown encodings	U27	\DeclareCharacterInheritance: new key 'inputenc'	
include LPLP	268	to set the input encoding	111
Font aliases: declare zeur and zeus (eulervm) as		\MT@rem@from@clist: model after \@removeelement	24
aliases of eur resp. eus (euler)	140	\MT@setup@: empty \MT@setup@ after use (compatibil-	
Inheritance: adapt to marvosym's changed encoding	146	ity with the combine class)	27
Protrusion: complete settings for Euler Fraktur and		\pickup@font: no tracing with trace package	92
Script fonts	195	\SetExpansion: new key: inputenc	105
fix: forgotten comma in mt-mvs.cfg; adapt to		\SetProtrusion: (et al.) new key: inputenc	104
marvosym's changed encoding	196		

2006/09/09 **Version 1.9f**

Protrusion: fix: euler-vm did not load euler settings	192	\MT@reset@context: only reset context if it has actu-	
\MT@curr@list@name: fix: \MessageBreak must not be		ally been changed	95
expanded	89	\MT@set@inh@list: fix: forgotten comma in the fea-	
\MT@gdef@n: new macros: global variants	19	tures list	112
\MT@get@inh@list: fix: input encoding must be set		\MT@set@named@keys: new macro: set name first, sim-	
after the inheritance list has been parsed	81	plify parsing of optional argument	106
\MT@glet: new macro	19	\SetProtrusion: (et al.) set catcodes before parsing	
		optional argument	104

2007/01/14 **Version 2.0**

General: compatibility with listings: set catcode of		<i>Miatidis</i>)	U8
backslash to zero (reported by <i>Steven Bath</i>) . . .	31	qualify hint about web documents with regard to	
compatibility with soul: register \textls and		older pdfTeX versions	U27
\lstyle	31	qualify hints about expansion error messages with	
new option: babel, by default false (language-		regard to older pdfTeX versions	U29
dependent setup suggested by <i>Ulrich Dirr</i>) . .	118	Font sets: new: footnotesize and scriptsize . . .	137
new option: letterspace, by default 100	120	new: smallcaps	137
new package letterspace: a stripped-down ver-		\DeclareMicrotypeBabelHook: new command: inter-	
sion, containing the letterspacing commands		action with babel	104
only	U1	\lstyle: fix: font switches don't pose a problem	
option 'babel': fix: switch off French babel's short-		anymore	69
hands properly (reported by <i>Daniel Flipo</i>) . . .	135	fix: letterspacing commands may be nested	69
option 'babel': switch off Turkish babel's short-		new command: letterspacing	69
hands	135	totally redone, using the new \letterspacefont	69
option 'unit', \SetProtrusion: deprecate value		\MT@declare@sets: fix: empty size list when redefin-	
'relative' completely	109	ing set	97
Documentation: add hint about how to increase		\MT@is@symbol: made even more robust	87
font_max and font_mem_size	U29	\MT@load@inputenc: sanitise catcodes before loading	
add hint about warning when tracking and expansion		input encoding (problem with listings)	49
is applied to a font	U30	\MT@pdftex@no: case 6: pdfTeX 1.40	14
add remark about 'disable' (previously draft)		\MT@setup@noligatures: maybe disable	
option disabling microtype (noted by <i>Michalis</i>		\MT@noligatures after the preamble	134

<code>\MT@split@name</code> : adjust to possible letterspacing ..	40	<code>\SetTracking</code> : new command: tracking	105
<code>\SetExtraKerning</code> : new command: additional kerning	106	<code>\textls</code> : new command: letterspacing	74
<code>\SetExtraSpacing</code> : new command: adjustment of interword spacing	106	starred version: remove spaces around text	74
		<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	7

2007/01/21 **Version 2.1**

General: compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i>)	32	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	69
fix: letterspace package forgot to load <code>keyval</code> ..	16	<code>\MT@orig@pickupfont</code> : compatibility with CJK: also check for its definition	91
<code>\slig</code> : new command: protect ligatures in letter-spaced text	69	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	74

2007/07/14 **Version 2.2**

General: disable microtype if <code>wordcount</code> is loaded (reported by <i>Ross Hetherington</i>)	27	<code>\MT@is@composite</code> : more robust: expand exactly once ..	88
new option: <code>copyfonts</code>	119	<code>\MT@is@symbol</code> : expand once more (for <code>frenchpro</code>) ..	87
simplify key declarations	108	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	65
use <code>catcode</code> trickery for <code>e-TeX</code> test	13	<code>\MT@lua</code> : (basic) support for <code>LuaTeX</code>	16
Documentation: add hint about error message with <code>pdfTeX 1.40</code>	U29	<code>\MT@pdfTeX@no</code> : case 7: <code>pdfTeX 1.40.4</code>	14
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i>)	U27	<code>\MT@preset@aux@space</code> : generalised	50
add overview	U4	<code>\MT@set@all@pr</code> : (et al.) allow empty values	43
logo transparency and amusement	U1	<code>\MT@set@inputenc</code> : only load <code>inputenc</code> files if necessary	48
Font aliases: declare <code>chr</code> (<code>chmath</code>) as an alias of <code>bch</code> (reported by <i>Geoff Vallis</i>)	140	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in <code>pdfTeX 1.40.4</code>) possibility to customise interword spacing	66
declare <code>fp9x</code> , <code>fp9j</code> (FPL Neu) as aliases of <code>pp1[xj]</code> ..	139	<code>\MT@setup@expansion</code> : warning if stretch or shrink aren't multiples of step	131
Font sets: default set for tracking: <code>smallcaps</code>	138	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore . only add features that are available with the respective <code>pdfTeX</code>	37
Inheritance: remove <code>'</code> → <code>'127</code>	142	<code>\MT@setupfont@hook</code> : restore percent character if Galician <code>babel</code> is loaded	29
Protrusion: settings for Bitstream Letter Gothic ..	150	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	44
Spacing: add sample	197	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	64
Tracking: add ligatures that are to be disabled ..	147	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled ..	64
<code>\DeclareMicrotypeVariants</code> : new command	102	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	133
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	104	<code>\MT@with@babel@and@T</code> : also inspect class options ..	28
<code>\slig</code> : always defined	69	<code>\pickupfont</code> : letterspace: setup inside group ..	92
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	42	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	105
<code>\MT@copy@font</code> : optionally work on copies of fonts ..	38	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i>)	105
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	79	third argument may be empty	105
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	46	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	95
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	69		
<code>\MT@ifdimen</code> : employ <code>LuaTeX</code> features if available ..	21		
<code>\MT@ifint</code> : employ <code>LuaTeX</code> features if available ..	21		
<code>\MT@ifstreq</code> : employ <code>LuaTeX</code> features if available ..	22		
fix: <code>ε-TeX</code> version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i>)	22		

2007/12/23 **Version 2.3**

General: disable <code>\microtypecontext</code> in <code>hyperref's</code> <code>\pdfstringdef</code>	31	Documentation: add kerning sample	U18
fix: really switch off Turkish shorthands	135	add letterspacing illustration	U16
new value for verbose option: <code>silent</code> (suggested by <i>Karl Berry</i>)	120	<code>\do@subst@correct</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i>)	93
turned some warnings into errors	120	<code>\slig</code> : redone: extract outer kerns from current letterspacing amount	69

<code>\microtypecontext</code> : made robust (reported by <i>Stephan Hennig</i>)	95	<code>\MT@set@curr@os</code> : adjusting spaces made more reliable	66
<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code>	78	<code>\MT@set@tr@codes</code> : also adjust tracking if protrusion is not enabled, and even for <code>letterspace</code> (reported by <i>Stephan Hennig</i>)	66
<code>\MT@define@set@key@font</code> : font: single asterisk means normal font	100	possibility to customise outer kerning (suggested by <i>Stephan Hennig</i>)	66
<code>\MT@in@clist</code> : don't use <code>\x</code> (reported by <i>Peter Meier</i>)	24	<code>\MT@SetTracking</code> : sanity check for value	105
<code>\MT@is@active</code> : support for extended Unicode (<code>inputenc/utf8x</code> resp. <code>ucs</code>) – experimental	85	<code>\MT@setup@tracking</code> : enable protrusion when tracking is enabled	132
<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in v2.2)	75	<code>\MT@tr@outer@l</code> : only change pre outer space if it contains <code>shrink</code>	71
<code>\MT@plain</code> : <code>letterspace</code> : support for <code>explain/miniltx</code>	12		

2008/02/29 **Version 2.3a**

General: fix test for <code>soul</code> under plain <code>T_EX</code>	31	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the <code>letterspace</code> package	5
Documentation: add hint about <code>babel</code> having to be loaded first	U28	<code>\MT@getkey</code> : fix: <code>key=val</code> in class options list	127
add table of available and enabled features	U6	<code>\MT@set@codes</code> : generalised	44
Protrusion: adjust LMR quotation marks again	155	<code>\MT@setupfont@hook</code> : restore percent character if Mexican <code>babel</code> is loaded	29
<code>\MT@error@doesnt@work</code> : error messages if <code>pdfT_EX</code> is too old for extensions	133		

2008/06/04 **Version 2.3b**

<code>\MT@exp@gcs</code> : new macro: reduce save stack size	19	also check for its definition	91
<code>\MT@font@copy</code> : enable font copies also with protrusion contexts (reported by <i>Nathan Rosenblum</i>)	38	<code>\MT@requires@l_{at}ex</code> : new macro	13
<code>\MT@get@size@%</code> : grouping	100	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i>)	66
<code>\MT@noligatures@%</code> : fix: warning messages for unknown slots	76	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode	71
<code>\MT@orig@pickupfont</code> : compatibility with <code>CJKut_f8</code> :		make <code>\spaceskip-aware</code> (<code>ragged2e</code>)	71
		<code>\MT@tr@outer@r@%</code> : additional test for horizontal mode	72

2008/11/11 **Version 2.3c**

General: <code>LuaT_EX</code> supported by default	15	coding (reported by <i>Vasile Gaburici</i>)	144
Documentation: add hint about spacing being experimental	U27	<code>\MT@detokenize@c</code> : fix: remove last space only (reported by <i>Ulrich Dirr</i>)	20
add hint about partial incompatibility with <code>CJK</code>	U28	<code>\MT@tr@outer@r@%</code> : additional test for horizontal mode (reported by <i>Sveinung Heggen</i>)	72
Inheritance: add <code>\textcommabelow[STst]</code> to <code>QX</code> en-			

2009/03/27 **Version 2.3d**

General: fix pinyin compatibility check (reported by <i>Silas S. Brown</i>)	32	(reported by <i>Ulrich Dirr</i>)	70
move setup to the very end (for <i>Colin Rourke</i>)	136	<code>\MT@setup@expansion</code> : default step: 1 for <code>pdfT_EX</code> versions ≥ 1.40	129
<code>\ifMT@inannot</code> : use <code>pdf_texcmds</code> for debugging	7	<code>\MT@tr@outer@r@%</code> : don't use <code>\x</code> (reported by <i>Ulrich Dirr</i>)	72
<code>\l_sstyle</code> : disable for <code>LuaT_EX</code>	69	fix: don't adjust in math mode (reported by <i>Christoph Bier</i>)	72
make invalid in math mode	69	fix: don't adjust inside discretionary (reported by <i>Maverick Woo</i>)	72
<code>\microtypesetup</code> : select font after setup	124	<code>\MT@tr@set@okern</code> : allow empty value for outer kerning	74
<code>\MT@check@active@set</code> : warning for missing default sets	123	<code>\textls</code> : make math mode aware	74
<code>\MT@lua</code> : update for <code>LuaT_EX 0.36</code>	16		
<code>\MT@set@tr@codes</code> : allow zero tracking	65		
<code>\MT@set@tr@zero</code> : fix: allow switching off tracking			

2009/11/09 **Version 2.3e**

Expansion: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	147	add T2A encoding	137
Font sets: <code>sc*</code> instead of <code>sc</code> in <code>smallcaps</code> set	137	Protrusion: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	156

Spacing: settings for T2A encoding (contributed by <i>Karl Karlsson</i>)	198	<i>Marcin Borkowski</i>	27
<code>\MT@fix@fontdimen@six</code> : fix: gobbling settings with tracking failed (reported by <i>Leo</i>)	39	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (first reported by <i>Christian Stark</i>)	29
<code>\MT@setup@</code> : make space-unaware (requested by		<code>\MT@tr@outer@r@</code> : fix: set current kerning and spacing again (found by <i>Lars Rönnbäck</i>)	72
2010/01/10	Version 2.4		
General: new file <code>microtype.lua</code> containing the lua functions (contributed by <i>Élie Roux</i>)	18	Protrusion: settings for T2A encoded Minion (contributed by <i>Karl Karlsson</i>)	156
2013/03/13	Version 2.5		
General: allow contexts for LuaTeX	108	<code>\MT@define@code@key@family</code> : compatibility with <code>fontspec</code> : remove its internal counter (reported by <i>Till A. Heilmann</i>)	107
disable ‘DVIPoutput’ option for XeTeX	119	<code>\MT@define@code@key@font</code> : scrub <code>fontspec</code> feature count (found by <i>Meho R</i>)	107
fix: check whether ‘(file)/(line)’ list name already exists (reported by <i>Till A. Heilmann</i>)	108	<code>\MT@do@font</code> : adapt for LuaTeX	25
letterspacing with LuaTeX 0.62	64	adapt for XeTeX	26
new files: <code>microtype-pdftex.def</code> , <code>microtype-xetex.def</code> , <code>microtype-luatex.def</code> , containing engine-specific definitions	13	<code>\MT@get@slot@</code> : adapt for LuaTeX (requested by <i>Georg Duffner</i>)	82
protrusion with XeTeX	15	adapt for XeTeX	82
restore <code>\space</code> inside listings (reported by <i>Rolf Dieterich</i>)	31	<code>\MT@if@outer@next</code> : fix: conflict with <code>amsmath</code> (reported by <i>Scott Pakin</i>)	72
Documentation: add hint about LuaTeX compatibility	U28	<code>\MT@info@missing@char</code> : fix error message for XeTeX (reported by <i>Juan Acevedo</i>)	46
add hint about spacing and <code>ragged2e</code>	U27	<code>\MT@is@charx</code> : compatibility with <code>xunicode</code>	88
add hint about dtx source code	U30	<code>\MT@ledmac@setup</code> : fix to work with XeTeX (reported by <i>Maïeul Rouquette</i>)	28
include <code>microtype-logo.dtx</code> and <code>microtype-lssample.dtx</code>	234	<code>\MT@l@s@set@l@s</code> : allow formulas in optional argument to <code>\textls</code> (fix by <i>Heiko Oberdiek</i>)	74
Font aliases: declare <code>lmsy</code> and <code>lmm</code> as aliases of <code>cmsy</code> resp. <code>cmm</code> (reported by <i>Jonas Hogstrom</i>)	138	<code>\MT@microtypecontext</code> : fix: ensure to set up math fonts (reported by <i>RazorXsr</i>)	95
declare <code>zgmxc</code> etc. (<code>garamondx</code>) as aliases of <code>ugm</code>	140	<code>\MT@register@subst@font</code> : only register substituted font if it isn’t registered already (reported by <i>George Gratzner</i> and <i>Josep Maria Font</i>)	93
declare Latin Modern Roman (OpenType version) as alias of <code>lmr</code> when <code>fontspec</code> is loaded	138	<code>\MT@register@subst@font@cx</code> : only register if it isn’t registered already	94
declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version)	139	<code>\MT@scrubfeatures</code> : compatibility with <code>fontspec</code> : remove its internal counter	40
Font sets: add EU1 and EU2 encodings	137	<code>\MT@set@all@pr</code> : fix: remove space (found by <i>Meho R</i>)	43
Inheritance: add rudimentary list for EU1 and EU2	145	<code>\MT@set@pr@codes</code> : make info about generic settings encoding-specific (reported by <i>Sebastian Schubert</i>)	43
Protrusion: add default lists for EU1 and EU2	154	<code>\MT@setup@spacing</code> : warning with <code>ragged2e</code> (reported by <i>Steffen Hoffmann</i>)	132
improvements to Computer Modern Roman italics (contributed by <i>Hendrik Vogt</i>)	160	<code>\MT@setupfont</code> : select font with <code>fontspec</code> (found by <i>Georg Duffner</i>)	37
Tracking: add EU2 encoding to default list	147	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>mathastext</code> is loaded (found by <i>Seamus Bradley</i>)	30
<code>\DeclareCharacterInheritance</code> : allow more than one encoding	111		
<code>\DeclareMicrotypeAlias</code> : ignore spaces	103		
<code>\ifMT@nofamily</code> : info if settings are not family-specific (suggested by <i>Hàn Thê Thành</i>)	42		
<code>\LoadMicrotypeFile</code> : remove all spaces in font name	103		
<code>\lssstyle</code> : fix: ensure to set up math fonts (reported by <i>RazorXsr</i>)	69		
2013/05/23	Version 2.5a		
General: use <code>luatexbase</code> instead of <code>luatextra</code> (contributed by <i>Élie Roux</i>)	18	tributed by <i>Élie Roux</i>)	84
Documentation: add notes on typesetting the documentation	U30	<code>\MT@led@unhbox@line</code> : simplified	28
include OpenType configuration files	202	<code>\MT@ledmac@setup</code> : support for <code>eledmac</code>	28
<code>\MT@afteraftergroup</code> : fix: get outer kerning and spacing of nested letterspacing right	67	<code>\MT@l@s@outer@k</code> : add marker for tightly nested letterspacing	75
<code>\MT@get@slot@</code> : adapt to <code>luaotfload v2.2</code> (con-		<code>\MT@set@tr@codes</code> : fix: load font for <code>fontspec</code>	66
		<code>\MT@xspace</code> : fix outer spacing problem with <code>xspace</code> (reported by <i>Dave</i>)	73

2016/05/01 **Version 2.6**

General: load luaotfload with LuaTeX	18	\MT@engine: fix test with LuaTeX 0.85	13
redefine \MT@setupfont@hook globally for problem with tikzposter (reported by <i>Sam Mason</i>)	30	\MT@get@slot@: fix: could fail with XeTeX (reported by <i>Christopher Schramm</i>)	83
Documentation: add hint about partial incompatibility with xeCJK and luatexja	U28	\MT@is@xchar: update for fontspec's TU encoding	88
missing characters printed with Charis SIL	202	\MT@ledmac@setup: support for reledmac	28
Font sets: add TU encoding (notified by <i>Will Robertson</i>)	137	\MT@luatex@no: update for LuaTeX 0.85 (renamed primitives)	15
add si and scit to smallcaps set (reported by <i>uli</i>)	137	\MT@noligatures@: use luaotfload function to keep/inhibit ligatures	76
new: allmath-nott and alltext-nott (suggested by <i>Karl Berry</i>)	137	\MT@orig@pickupfont: (in)compatibility with luatexja: disable unknown slots warnings (reported by <i>Max</i>)	91
Inheritance: add TU encoding	145	(in)compatibility with xeCJK: disable unknown slots warnings (reported by <i>HcN</i>)	91
Protrusion: add TU encoding to lists	154	compatibility with xeCJK: pretend that CJK wasn't loaded	91
Tracking: add TU encoding to default list	147	\MT@set@tr@codes: use luaotfload's kernfactor feature if available	65
\DeclareMicrotypeSet: ignore spaces	96	\MT@xspace: fix outer spacing problem with (not only) algorithm (reported by <i>Henning and Ronnie Marksch</i>)	73
\DeclareMicrotypeSetDefault: ignore spaces	102	\UseMicrotypeSet: ignore spaces	101
\DeclareMicrotypeVariants: ignore spaces	102		
\sstyle: fix: ensure to set up math fonts (reported by <i>kleenstar</i>)	69		
\microtypecontext: allow activate shortcut (reported by <i>Karl Berry</i>)	95		
\MT@declare@sets: fix: undefine lists for redefining	97		
\MT@do@font: speed up for LuaTeX	25		

2016/05/14 **Version 2.6a**

General: fixes for letterspace package with LuaTeX	25	Voß)	25
\MT@do@font: fix lua function (reported by <i>Herbert</i>		\MT@ls@fontspec@font: fix for value of ± 1000	67

2017/07/07 **Version 2.7**

General: drop luatexbase with recent L ^A TeX	18	\MT@check@range@: don't warn for override if conflicting list is loaded	116
warning with minimal class	27	\MT@is@composite: compatibility with L ^A TeX 2017/01/01 (\DeclareUnicodeComposite) (reported by <i>Ulrike Fischer</i> and <i>jcr</i>)	89
Documentation: mention that additional kerning does not work in math mode (discovered by <i>Daniel</i>)	U17	\MT@ls@fontspec@font: fix for 'file:font' spec (reported by <i>Reinhard Kotucha</i>)	67
Font aliases: declare aliases for newpx	139	\MT@permute@@@@@: don't warn for override if conflicting list is loaded	115
declare aliases for newtx	139	\MT@reset@ef@codes: only reset \efcodes for older LuaTeX versions	60
declare aliases for tempora	139	\MT@setup@expansion: don't disable automatic expansion for DVI output with LuaTeX	130
declare aliases for XCharter	140	\MT@tikz@setup: compatibility with tikz (again)	29
declare Latin Modern Roman as alias of lmr with new L ^A TeX format (reported by <i>Ulrike Fischer</i>)	138	\MT@warn@tracking@DVI: don't warn for letterspacing in DVI mode with LuaTeX	133
Protrusion: automatically choose correct names for Charis SIL small caps (reported by <i>ltcomdata</i>)	223		
\sstyle: fix: prevent infinite loop with psnfss and exscale packages (reported by <i>user11126</i> , solution by <i>Ulrike Fischer</i>)	69		

2018/01/14 **Version 2.7a**

General: disallow non-automatic expansion with LuaTeX	110	\MT@get@highlevel: test whether \...default is defined	98
\MT@auto: remove 'autoexpand' for LuaTeX 1.0.6 (reported by <i>Ulrike Fischer</i>)	129	\MT@get@slot: expand active characters earlier	81
with LuaTeX, non-automatic font expansion is no longer possible (as confirmed by <i>Hans Hagen</i>)	129	\MT@info@nottracking@: defer 'No tracking' message	41
		\MT@is@active: compatibility with newunicodechar (reported by <i>Nils Anders Danielsson</i>)	86

2019/02/28 **Version 2.7b**

General: update lua function <code>microtype.info</code> after changes in <code>luaotfload</code> (reported by <i>Moritz Wemheuer</i> and <i>Ulrike Fischer</i>)	18	(reported by <i>Franz Wexler</i>)	143
Documentation: update hint about non-7-bit characters (notified by <i>Frank Mittelbach</i>)	U28	<code>\MT@info@missing@char</code> : fix message for glyphs specified as names in $X_{\text{q}}\text{T}_{\text{E}}\text{X}$ (reported by <i>Paolo Ney</i>)	46
Inheritance: add <code>textquotedblleft</code> ligature to OT4		<code>\MT@setupfont</code> : always select current font with $X_{\text{q}}\text{T}_{\text{E}}\text{X}$ and $\text{LuaT}_{\text{E}}\text{X}$ (reported by <i>Paolo Ney</i> , solution by <i>Ulrike Fischer</i>)	37

2019/10/10 **Version 2.7c**

General: turn warning into info when overwriting the <code>keepligature</code> function (reported by <i>Andy N</i>)	76	<code>\MT@is@symbol</code> : take care of <code>\remove@tlig</code>	87
<code>\MT@is@active</code> : compatibility with $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2019/10/01	86	<code>\showhyphens</code> : compatibility with $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2019/10/01 (reported by <i>Phelype Oleinik</i> and <i>Falk Hanisch</i>)	131

2019/11/18 **Version 2.7d**

<code>\MT@copy@font@:</code> in $\text{LuaT}_{\text{E}}\text{X}$, don't use the <code>\copyfont</code> primitive, but load the font anew (reported by <i>Paolo Polesana</i> and <i>Oliver Kopp</i>)	38	from list (reported by <i>Markus Kohm</i>)	93
<code>\MT@register@subst@font</code> : remove substitute font		<code>\MT@register@subst@font@cx</code> : remove substitute font from lists	94

2020/12/07 **Version 2.8**

General: <code>letterspace</code> works with $\text{eT}_{\text{E}}\text{X}$ only	13	<code>\lstyle</code> : fix: enforce math setup, again	69
compatibility with <code>soul</code> : patch for font change (reported by <i>Md Ayquassar</i>)	31	<code>\microtypecontext</code> : fix activate shortcut	95
fix for <code>luaotfbase</code>	18	ignore spaces	95
Documentation: declare <code>DVIoutput</code> option deprecated	U8	<code>\MT@do@font</code> : fix for $X_{\text{q}}\text{T}_{\text{E}}\text{X}$	26
squash fake news about automatic font expansion with <code>dvilualatex</code>	U7	simplify lua function	25
Font aliases: declare aliases for <code>step</code> and <code>domitian</code> (notified by <i>Daniel Benjamin Miller</i>)	139	<code>\MT@fix@fontdimen@six</code> : try to fix zero <code>\fontdimen 6</code>	39
declare aliases for <code>stix</code> and <code>stix2</code> fonts	139	<code>\MT@if@luaotf@font</code> : use lua function	25
declare New Computer Modern as an alias of Latin Modern Roman	139	<code>\MT@if@streq</code> : use $X_{\text{q}}\text{T}_{\text{E}}\text{X}$'s <code>\stringcmp</code>	22
Font sets: default set for expansion: <code>alltext-nott</code> (suggested by <i>Aman Mehra</i>)	138	<code>\MT@setup@expansion</code> : warning when expanding in DVI mode with $\text{LuaT}_{\text{E}}\text{X}$ (reported by <i>Daniel Benjamin Miller</i>)	128
default set for spacing: <code>alltext-nott</code>	138	<code>\MT@tr@set@space@:</code> simplified	71
		<code>\MT@tr@unit@:</code> fix: allow unit regardless whether letterspacing is set	68
		<code>\textmicrotypecontext</code> : ignore spaces	95

2021/02/22 **Version 2.8a**

General: rename <code>mt-pad.cfg</code> to <code>mt-EBGaramond.cfg</code> (requested by <i>Karl Berry</i>)	137	Inheritance: specify 'ff' ligature as Unicode instead of glyph name	202
rename <code>mt-PalatinoLinotype.cfg</code> to <code>mt-Palatino.cfg</code> (requested by <i>Karl Berry</i>)	202	Protrusion: hide <code>euorotc</code> settings (requested by <i>Karl Berry</i>)	196
Documentation: reorganise table 1 for clarification (suggested by <i>Daniel Benjamin Miller</i>)	U6	Spacing: add dummy settings (reported by <i>DOR-papst</i>)	200
Font aliases: add <code>-LF</code> , <code>-TLF</code> , <code>-0sF</code> and <code>-T0sF</code> as variants	138	<code>\DeclareCharacterInheritance</code> : skip settings for tracking	111
declare <code>mmodern</code> fonts as aliases of Latin Modern (reported by <i>Daniel Benjamin Miller</i>)	138	<code>\MT@get@size@:</code> compatibility with <code>svjour3</code> (reported by <i>Ekkehart Schlicht</i> (no relation!))	100
reference New Computer Modern also by file name (reported by <i>Canageek</i>)	139	<code>\MT@gl@et</code> : use $\text{LuaT}_{\text{E}}\text{X}$'s <code>\gl@et</code> , if available	19

2021/02/25 **Version 2.8b**

Inheritance: dummy settings for the Font Awesome font (<code>mt-FontAwesome.cfg</code>)	210	settings for the Lato font (<code>mt-Lato.cfg</code>) (reported by <i>dsedivec</i>)	210
---	-----	--	-----

2021/03/14 **Version 2.8c**

<code>\ifMT@fontspec</code> : add to hook for fontspec	29	<code>\MT@remove@tlig</code> : fix for text commands containing conditionals (reported by <i>Frank Mittelbach</i>) . . .	88
<code>\MT@is@opt@char</code> : fix for optionally defined glyphs (reported by <i>Frank Mittelbach</i>)	87		

2021/10/31 **Version 3.0**

General: letterspace loads <code>microtype.lua</code>	18	Protrusion: LGR settings for EB Garamond	159
compatibility fix for unicode-math with <code>luainputenc</code> (reported by <i>tnull</i>)	32	fix quotation marks in LMR	211
move development to GitHub	U1	settings for EB Garamond (OpenType)	224
new option: <code>disable</code> , replacing the <code>draft</code> option (suggested by <i>Frank Mittelbach</i>)	120	settings for EB Garamond figure variants	154
new options: <code>patch</code> and <code>nopatch</code>	121	settings for EB Garamond small caps italics	178
new package: <code>microtype-show</code> for visual debugging	8	settings for New Computer Modern (provided by <i>Antonis Tsolomitis</i>)	211
protrusion patch for TOC	34	<code>\ifMT@fontspec</code> : fix for changed hook name order (reported by <i>Frank Mittelbach</i> and <i>Karl Berry</i>)	29
protrusion patch for <code>\item</code>	34	<code>\leftprotrusion</code> : new command	51
protrusion patch for equation numbers (provided by <i>Holger Gerhardt</i>)	35	<code>\MT@do@font</code> : guard against return values outside Unicode range (for <i>HarfBuzz</i>)	25
protrusion patch for footnote text	35	<code>\MT@get@slot@</code> : convert floats to integers (reported by <i>azur</i>)	84
remove option <code>final</code>	120	guard against return values outside Unicode range (for <i>HarfBuzz</i>)	84
require package <code>etoolbox</code> (for <code>patch</code> option)	16	<code>\MT@has@inh@prefix</code> : new: automatical protrusion based on char widths difference (suggested by <i>Daniel Benjamin Miller</i>)	113
Documentation: split up in User manual (<code>microtype.pdf</code>) and Implementation doc (<code>microtype-code.pdf</code>)	U1	<code>\MT@is@opt@char</code> : fix for incompatibility with syntax (reported by <i>Safron</i>)	87
add hint about unknown slot numbers	U28	<code>\MT@ls@fontspec@font</code> : use <code>lua</code> to parse font spec/fix for font spec containing spaces (reported by <i>Aman Mehra</i>)	67
remove hint about extra TOC leader dot (fixed in \LaTeX)	U27	<code>\MT@old@cmd</code> : old command names will raise an error	6
remove hint about web documents with pre-1.40 <code>pdfTeX</code> (it's been 14 years . . .)	U27	<code>\MT@pdf@tex@no</code> : case 8: <code>pdfTeX 1.40.23</code>	14
rename option <code>draft</code> to <code>disable</code> (insistently requested by <i>Frank Mittelbach</i>)	U8	<code>\MT@set@inh@list</code> : fix: grouping with <code>\begin@group . . . \end@group</code> instead of <code>{. . .}</code> (reported by <i>chsk</i>)	111
split up table of fonts with tailored protrusion settings	U22	<code>\MT@set@tr@codes</code> : with <code>LuaTeX</code> , tracking also works for fonts with a zero <code>\fontdimen 6</code>	65
Font aliases: declare basic alias for Bergamo Std	140	with <code>pdfTeX 1.40.23</code> , tracking also works for fonts with a zero <code>\fontdimen 6</code>	65
declare basic aliases for the Fontin font	140	<code>\noprotrusion</code> : provide command for older \LaTeX versions	51
declare empty aliases for <code>fontawesome5</code>	140	<code>\rightprotrusion</code> : new command	51
New Computer Modern has its own settings	139		
Inheritance: add settings for LGR	146		
generalise basic inheritance settings	210		
generalise empty settings	210		
settings for EB Garamond (OpenType)	206		
settings for New Computer Modern (provided by <i>Antonis Tsolomitis</i>)	202		

2021/12/02 **Version 3.0a**

General: abort earlier if no capable engine found	16	<code>\MT@apply@patch</code> : compatibility with <code>babel/spanish</code> : fix catcodes	33
allow <code>\microtypesetup{(no)patch=all}</code>	126	<code>\MT@get@prot</code> : compatibility fix for <code>csquotes</code> (reported by <i>Gustavo Barros</i>) [issue #1]	51
define function <code>math.tointeger</code> for older <code>LuaTeX</code> versions	19	reset counters	51
fix protrusion patch footnote with KOMA classes (reported by <i>MisterFiLou</i>) [issue #2]	36	set penalties to zero and <code>vfuzz</code> to max (reported by <i>florian</i>)	51
fix protrusion patch footnote with <code>hyperref</code> (reported by <i>Liang-Bo Wang</i>) [issue #2]	35	<code>\MT@prot@addgroup</code> : rescanned grouped material to allow verbatim etc. (reported by <i>Torsten Schuetze</i>)	53
protrusion patch <code>eqnum</code> for <code>IEEEtran</code>	35	<code>\MT@redefine@patch</code> : facilitate redefinition of patches	33
Font aliases: declare CMU Serif as an alias of New Computer Modern	139	<code>\MT@with@babel@and@T</code> : fix grouping	28

2021/12/10	Version 3.0b		
	General: disable patches for tex4ht	31	by Akira Yokosawa and <i>theufman</i> [issues #3,#4] 53
	\MT@get@prot: unconditionally \leavevmode	51	\MT@prot@l@: make \long again (reported by Akira Yokosawa) [issue #3] 51
	\MT@prot@addgroup: don't rescan anymore (reported		
2022/02/22	Version 3.0c		
	General: info that protrusion patch eqnum may not be effective with mathtools (reported by <i>user182849</i>)	35	\MT@noindent: use \RawNoindent, if available [issue #8] 52
	make babel option work with polyglossia	135	\MT@prot@iffirstcmd: use \long variant of \@car (reported by <i>frafl</i>) [issue #6] 53
	protrusion patch footnote: also for minipages	35	\MT@prot@l@: next try at removing \long 51
	\MT@csq@eggroup: compatibility with csquotes, again (reported by <i>NightShade</i>)	52	\MT@prot@r: no longer \long 51
			\noprotrusionifhmode: new command 51
2022/03/14	Version 3.0d		
	\lslig: define \font@name (reported by <i>Ulrike Fischer</i>) [issue #12]	69	<i>Brian Dunn</i> [issue #14] 51
	\MT@get@prot: no longer reset counters (reported by		\MT@prot@addgroup: no brace delimiting (reported by <i>Nelson Lago</i>) [issue #10] 53
2022/06/20	Version 3.0e		
	General: disable csquotes's grouping control for \leftprotrusion (reported by <i>Ralf Steinle</i> and <i>Denis Bitouzé</i>)	30	\MT@is@active@hook: fix for doc/shortvrb's \MakeShortVerb 86
	fix protrusion patch footnote for hyperref if \hyper@nopatch@footnote is defined	35	fix for listings's \lstMakeShortInline (reported by <i>Denis Bitouzé</i>) 86
	fix protrusion patch footnote for hyperref with option <code>implicit=false</code>	35	hook for active chars 86
	fix protrusion patch footnote with beamer	36	\MT@prot@addgroup: only add group if very first token 53
	protrusion patch eqnum: adjust to robustness of \qref (L ^A T _E X 2022-06-01), cf. [issue #19]	35	\MT@prot@check@cmds: add standard font selection commands 55
	Font aliases: declare aliases for stickstoo fonts	139	ignore \@empty 55
	\MT@csq@eggroup: add \relax (for csquotes)	52	new macro: make list of commands extensible 55
	\MT@get@prot: revert \leavevmode	51	\MT@prot@get@first@token: don't gobble previously captured content 57
			\MT@prot@ifmacro@: use etoolbox facilities 53
2022/06/23	Version 3.0f		
	\MT@prot@check@cmds: add new NFSS commands	56	codings (also fixes [issue #20], reported by <i>Christophe Dervieux</i>) 53
	add microtype's commands	55	
	\MT@prot@iflicrmd: no need to know about en-		
2023/03/06	Version 3.1		
	General: fix test for KOMA classes in protrusion patch footnote (also reported by <i>David Purton</i>) [issue #26]	36	\MT@prot@check@cmds: add ltxdoc and doc abbreviations 56
	new protrusion patch <code>verbatim</code>	36	add \em 55
	Documentation: add hint about protrusion patch warnings	U29	add \MakeUppercase and friends 55
	\DeclareMicrotypeFilePrefix: new command (suggested by <i>rallg</i>) [issue #28]	103	add csquotes's commands (reported by <i>Shen Zhou Hong</i>) [issue #25] 57
	\MT@active@features: always activate contexts for tracking	94	add fontaxes commands 56
	\MT@define@patch: check if patch defined	32	add nfssect-cfr commands 56
	define patches globally	32	\MT@prot@check@E: new macro: take care of commands that enclose their argument 54
	\MT@find@file: simplify	77	\MT@prot@check@e: new macro: take care of starred commands that enclose their argument 54
	\MT@prot@addgroup: allow multiple groups	53	\MT@prot@check@eX: new macro: replace enclosing commands 54
	\MT@prot@check: allow replacement command	53	\MT@prot@check@F: new macro: for text commands 55

<code>\MT@prot@check@l</code> : new macro: for commands that enclose their second argument	54	switching enclosing commands	55
<code>\MT@prot@check@lX</code> : new macro: replace language-		<code>\MT@prot@get@first@token</code> : expand toks once	57
		<code>\MT@prot@l@tc</code> : new macro: for text commands . . .	51

2023/03/13 **Version 3.1a**

<code>\MT@prot@check@F</code> : fix: execute (and empty) <code>\MT@toks</code> (reported by <i>Artur A. Marczok</i> and <i>Uwe Siart</i>) . . .	55
---	----

2024/03/29 **Version 3.1b**

General: add <code>DEPENDS.txt</code> (suggested by <i>Bruno Vic- tal</i>)	U1	with an optional argument (after <i>scholnik</i> re- ported that <code>\MakeUppercase</code> and friends support one) [issue #29]	55
<code>\MT@check@MT@version</code> : check if <code>.sty</code> and <code>.def</code> (and <code>.lua</code>) file versions match	122	<code>\MT@prot@check@F</code> : fix: failed when group began with another command [issue #31]	55
<code>\MT@fix@fontdimen@six</code> : compatibility with LuaTeX in DVI mode (reported by <i>Linas Stonys</i>)	39	<code>\MT@with@babel@and@T</code> : don't use <code>\@pkgextension</code> (was <code>\@onlypreamble</code> in older L ^A T _E X versions) . . .	28
<code>\MT@i f@expanding@F</code> : new macro: check whether in- side expanding context	20	<code>\pickup@font</code> : guard against expanding context . . .	92
<code>\MT@prot@check@C</code> : new macro: for text commands			

D Index

Numbers in upright shape refer to the *page* where the corresponding entry is described (bold face) resp. occurs. Numbers in italic shape refer to the *code line* where the corresponding entry is defined (underlined) resp. used. Numbers prefixed with ‘U’ refer to the User manual.

Options	<code>DVIoutput</code>	U8	<code>patch</code>	U6
	<code>activate</code>	U5	<code>protrusion</code>	U5
	<code>auto</code>	U7	<code>selected</code>	U7
	<code>babel</code>	U8	<code>shrink</code>	U7
	<code>config</code>	U8	<code>spacing</code>	U6
	<code>disable</code>	U8	<code>step</code>	U7
	<code>expansion</code>	U5	<code>stretch</code>	U7
	<code>factor</code>	U6	<code>tracking</code>	U5
	<code>kerning</code>	U6	<code>unit</code>	U7
	<code>letterspace</code>	U8	<code>verbose</code>	U8
	<code>nopatch</code>	U6		
Commands	<code>\DeclareCharacterInheritance</code>	U19	<code>\SetProtrusion</code>	U12
	<code>\DeclareMicrotypeAlias</code>	U20	<code>\SetTracking</code>	U15
	<code>\DeclareMicrotypeBabelHook</code>	U23	<code>\UseMicrotypeSet</code>	U10
	<code>\DeclareMicrotypeFilePrefix</code>	U21	<code>\leftprotrusion</code>	U25
	<code>\DeclareMicrotypeSet*</code>	U9	<code>\lslig</code>	U23
	<code>\DeclareMicrotypeSet</code>	U9	<code>\lstyle</code>	U23
	<code>\DeclareMicrotypeSetDefault</code>	U10	<code>\microtypecontext</code>	U21
	<code>\DeclareMicrotypeVariants*</code>	U20	<code>\microtypesetup</code>	U8
	<code>\DeclareMicrotypeVariants</code>	U20	<code>\noprotrusion</code>	U25
	<code>\DisableLigatures</code>	U24	<code>\noprotrusionifhmode</code>	U25
	<code>\LoadMicrotypeFile</code>	U20	<code>\rightprotrusion</code>	U25
	<code>\SetExpansion</code>	U13	<code>\textls*</code>	U23
	<code>\SetExtraKerning</code>	U17	<code>\textls</code>	U23
	<code>\SetExtraSpacing</code>	U18	<code>\textmicrotypecontext</code>	U22
A	<code>a0poster</code> (class)	100	<code>amsmath</code> (package)	U26, 35, 251
	<i>Juan Acevedo</i> (contributor)	U31, 251	<code>amssymb</code> (package)	188
	<code>activate</code> (option)	U5, 118, 248	<code>array</code> (package)	U25
	<code>\add@accent</code>	3912	<i>Donald Arseneau</i> (contributor)	U31
	<code>\adjustspacing</code>	434, 5396	<code>article</code> (class)	U22
	<code>ae</code> (package)	U21, 138	<code>auto</code> (option)	U7, 118
	<code>algorithm</code> (package)	252	<i>Md Ayquassar</i> (contributor)	U31, 253
	<code>alltt</code> (package)	U25, 36	<i>azur</i> (contributor)	U31, 254

- B** *Andreas Bühmann* (contributor) [U30](#), 246
babel (option) . . . [U8](#), [U23](#), [U28](#), [U34](#), 118, 248, 255
babel (package) [U4](#),
[U16](#), [U23](#), [U24](#), [U28](#), 2, 29, 33, 78, 104,
134, 135, 141, 245, 246, 248, 249, 250, 254
Gustavo Barros (contributor) [U31](#), 254
Steven Bath (contributor) [U31](#), 248
beamer (class) [U25](#), 34, 36, 255
- C** *Canageek* (contributor) [U31](#), 253
David Carlisle (contributor) [U31](#)
\cf@encoding 1432
chemsym (package) 245
chmath (package) [U21](#), 140, 249
chsk (contributor) [U31](#), 254
CJK (package) [U28](#), 32, 91, 249, 250, 252
CJKutf8 (package) 91, 250
cm-super (package) [U7](#)
Joel Coffman (contributor) [U31](#)
- D** *Daniel* (contributor) [U31](#), 252
Nils Anders Danielsson (contributor) [U31](#), 252
Dave (contributor) [U31](#), 251
Loren B. Davis (contributor) [U30](#), 202
\DeclareCharacterInheritance [U19](#), [U35](#), 41, 112, 4761
\DeclareMicrotypeAlias [U20](#), 43, 77, 103, 4354
\DeclareMicrotypeBabelHook [U23](#), 46, 69, 4423
\DeclareMicrotypeFilePrefix [U21](#), [U32](#), 4391
\DeclareMicrotypeSet [U9](#),
[U10](#), [U11](#), 32, 62, 64, 78, 101, 104, 247, 4053
\DeclareMicrotypeSet* [U9](#), 4053
\DeclareMicrotypeSetDefault
. [U10](#), [U34](#), 34, 66, 123, 4309
\DeclareMicrotypeVariants [U20](#), [U33](#), 42, 63, 68, 4339
\DeclareMicrotypeVariants* [U20](#)
defersetup (option) 29, 118, 119, 124, 247
\define@newfont 3848, 3856, 3860, 3870
Christophe Dervieux (contributor) [U31](#), 255
- E** eco (package) [U21](#), 138
Maciej Eder (contributor) [U30](#), 144, 157, 248
\efcode 2441, 2467, 2468, 2509, 2511
eledmac (package) 28, 251
\encodingdefault 226
eplain (package) [U24](#), [U33](#), 4, 12, 28, 127, 250
e-TeX (engine) [U25](#), [U35](#), 13, 14, 16, 20, 26,
45, 53, 72, 100, 234, 245, 246, 247, 249, 253
etoolbox (package) 16, 33, 34, 53, 86, 254, 255
eucal (package) 193
- F** \f@encoding 1432
\f@family 3862, 3863
\f@size 2892, 3855, 3857, 3859, 3866, 3910
factor (option) [U6](#), [U12](#), [U35](#), 120, 121, 245
fancvrb (package) 36
final (option) 120, 245, 254
Ulrike Fischer (contributor) [U31](#), 252, 253, 255
fix-cm (package) 39
Daniel Flipo (contributor) [U31](#), 248
florian (contributor) [U31](#), 254
Josep Maria Font (contributor) [U31](#), 251
\font 214, 215,
233, 243, 255, 266, 269, 1419, 1495, 2720, 2880
\font@name 119, 245, 259, 1467, 1469, 1471,
1473, 1474, 1484, 1485, 2711, 2719, 2724,
2762, 2768, 2773, 2776, 2865, 2880, 2892,
2895, 2897, 2902, 2904, 2910, 2913, 2948,
2999, 3136, 3138, 3848, 3856, 3865, 3870,
3891, 3893, 3895, 3911, 3926, 3928, 3956, 3961
fontawesome (package) 140
fontawesome5 (package) 140, 254
fontaxes (package) 56, 255
\fontcharwd
279, 1770, 1774, 1780, 1980, 1981, 1989, 1990
\fontdimen 1499,
1500, 1506, 1509, 1806, 1811, 1911, 1914,
2737, 2738, 2768, 2895, 2941, 2942, 2949,
2950, 2957, 2959, 2965, 2980, 2999, 3149, 3152
\fontencoding 226, 2236
\fontfamily 2236
fontinst (package) 199, 234, 236
fontinstallationguide (package) 56, 177
\fontname 2719
\fontseries 226, 2236
\fontshape 226, 2236
- Axel Berger* (contributor) [U31](#), 198
Karl Berry (contributor) [U31](#), 249, 252, 253, 254
Christoph Bier (contributor) [U31](#), 245, 250
Denis Bitouzé (contributor) [U31](#), 255
Marcin Borkowski (contributor) [U31](#), 251
Seamus Bradley (contributor) [U31](#), 251
Silas S. Brown (contributor) [U31](#), 250
- color (package) 128
combine (class) 136, 248
config (option) [U8](#), [U20](#), [U34](#), 122, 246
contour (package) 128
copyfonts (option) 38, 118, 119, 249
crop (package) 128
csquotes (package) [U30](#), 29,
30, 51, 52, 54, 55, 57, 58, 153, 246, 254, 255
\curr@fontshape 2892, 3855, 3857, 3859, 3866, 3910
- Rolf Dieterich* (contributor) [U31](#), 251
Ulrich Durr (contributor) [U30](#), 197, 248, 250
disable (option) [U8](#), [U32](#), 120, 248, 254
\DisableLigatures
. [U6](#), [U24](#), [U33](#), [U34](#), 40, 67, 101, 134, 4396
\do@subst@correction 3909
doc (package) 56, 86, 255
docstrip (package) 4
domitian (package) [U21](#), 139, 253
DORpapst (contributor) [U31](#), 253
draft (option) 118, 120
dsevicec (contributor) [U31](#), 253
dsfont (package) 39
Georg Duffner (contributor) [U30](#), 251
Brian Dunn (contributor) [U31](#), 255
DVIoutput (option) [U8](#), 118, 119, 251, 253
Peter Dybala (contributor) [U31](#), 248
- eufrak (package) 195
euler (package) 89, 191, 248
eulerm (package) [U21](#), 140, 192, 248
euroitc (package) 196, 253
europs (package) 196
eurosans (package) 196
\expandglyphsinfont 2423
expansion (option) [U5](#), [U10](#), 117, 118, 128
exscale (package) 252

- `\fontsize` 2238
`fontspec` (package) [U5](#), [U25](#), [U28](#), [U32](#), [U33](#),
18, 40, 68, 88, 98, 107, 138, 145, 251, 252, 254
`fourier` (package) 39
- G** *Vasile Gaburici* (contributor) [U31](#), 250
`garamondx` (package) [U21](#), 140, 251
Bernard Gaulle (contributor) [U31](#), 245
Holger Gerhardt (contributor) [U31](#), 254
`german` (package) 5, 244
Philipp Gesang (contributor) [U30](#)
- H** *Morten Høgholm* (contributor) [U31](#), 249
Hans Hagen (contributor) 252
Falk Hanisch (contributor) [U31](#), 253
Harald Harders (contributor) [U30](#), 4, 150, 244
Steven E. Harris (contributor) [U31](#), 249
HcN (contributor) [U31](#), 252
Sveinung Heggen (contributor) [U31](#), 250
Till A. Heilmann (contributor) [U31](#), 251
Stephan Hennig (contributor) [U31](#), 246, 249, 250
Henning (contributor) [U31](#), 252
- I** `IEEEtran` (class) [U26](#), 35, 91, 254
`\iffontchar` 233, 243, 255,
988, 1817, 3706, 3716, 3718, 3784, 3796, 3797
`\ifMT@auto` [540](#), [5623](#), [5684](#)
`\ifMT@babel` [540](#), [5769](#), [5897](#)
`\ifMT@disable` 135, [540](#), [5516](#)
`\ifMT@do` 40, [1535](#), [1602](#), [2675](#), [3183](#)
`\ifMT@document` [582](#), [4910](#)
`\ifMT@draft` [540](#), [5175](#)
`\ifMT@engine@unfit` [402](#), [514](#)
`\ifMT@expansion` [540](#), [5572](#), [5586](#), [5681](#), [5727](#)
`\ifMT@fontspec` [1107](#), [1515](#), [4124](#), [4545](#), [4571](#)
`\ifMT@if@` [539](#), [1128](#), [1135](#),
[1160](#), [1200](#), [1388](#), [3685](#), [3952](#), [5017](#), [5917](#), [5920](#)
`\ifMT@inannot` [105](#)
`\ifMT@inlist@` [303](#), [319](#), [879](#),
[924](#), [1034](#), [1064](#), [1291](#), [1308](#), [1439](#), [1569](#),
[1586](#), [1595](#), [1613](#), [1631](#), [2673](#), [3297](#), [3308](#),
[3354](#), [3403](#), [3927](#), [3946](#), [3958](#), [4023](#), [4035](#), [4376](#)
`\ifMT@kerning` [540](#), [5783](#), [5910](#)
`\ifMT@nofamily` [1641](#), [1652](#)
- J** *Jcr* (contributor) [U31](#), 252
- K** *Karl Karlsson* (contributor)
..... [U30](#), 147, 150, 156, 198, 250, 251
Gabriel Kerneis (contributor) [U31](#)
`kerning` (option) [U6](#), [U10](#), [U33](#), 118
`keyval` (package) 16, 44, 141, 249
Tom Kink (contributor) [U31](#), 244
Hironori Kitagawa (contributor) [U31](#)
kleenstar (contributor) [U31](#), 252
`\knaccode` 2624, 2625, 2635, 2638, 2644
- L** *Nelson Lago* (contributor) [U31](#), 255
`ledmac` (package) [U30](#), [U34](#), 28, 91, 246
`ledpar` (package) [U30](#)
`\leftmargin kern` 1077
`\leftprotrusion` [U25](#), [U32](#),
30, 50, 51, 255, 1319, 1320, 1322, 1323,
1325, 1329, 1331, 1332, 1333, 1348, 1357,
1359, 1363, 1376, 1390, 1391, 1392, 1397,
1399, 1400, 1403, 1405, 1407, 1408, 1409, 2039
Philipp Lehman (contributor) [U30](#)
Leo (contributor) [U31](#), 251
`letterspace` (option)
- frafl* (contributor) [U31](#), 255
`french` (package) 246
`frenchpro` (package) 87, 245, 249
- `\glb@currsiz` 2868, 3995
`\glb@settings` 2869
`\GlyphScaleFactor` [U26](#), [149](#), 278, 292
`graphics` (package) 128
George Gratzner (contributor) [U31](#), 251
Gary L. Gray (contributor) [U31](#)
Ross Hetherington (contributor) [U31](#), 249
`hfoldsty` (package) [U21](#), 138
Steffen Hoffmann (contributor) [U31](#), 251
Jonas Hogstrom (contributor) [U31](#), 251
Shen Zhou Hong (contributor) [U31](#), 255
Michael Hoppe (contributor) [U31](#), 245, 247
hscm (contributor) [U31](#)
`hyperref` (package) [U25](#),
30, 31, 35, 91, 92, 128, 246, 247, 249, 254, 255
- `\ifMT@noligatures` [540](#), [5863](#)
`\ifMT@nonselected` [2403](#), [3416](#), [3424](#)
`\ifMT@norest` 3485, 3502, [3515](#), [3529](#), [3593](#)
`\ifMT@opt@auto` [5028](#), [5627](#), [5659](#)
`\ifMT@opt@DVI` [5028](#), [5541](#)
`\ifMT@opt@expansion` [5028](#), [5570](#), [5728](#)
`\ifMT@patch@ok` [1244](#), [1301](#), [1369](#)
`\ifMT@protrusion` [540](#), [1071](#), [1094](#), [5553](#), [5742](#)
`\ifMT@selected` [540](#), [5667](#), [5687](#)
`\ifMT@spacing` [540](#), [5751](#), [5768](#)
`\ifMT@tracking` [540](#), [5739](#), [5798](#)
`\ifMT@verbose` 245
`\ifMT@xunicode` [1107](#), [3731](#), [3788](#)
`ifpdf` (package) 91, 128
`\ifShowGlyphIndex` [U26](#), [149](#), [160](#)
`\ifShowMissingGlyphs` [U26](#), [149](#), [345](#), [1679](#)
`\iftracingmicrotypeinpdfall` [111](#)
`\ignoreligaturesinfont` 3187
`inputenc` (package) [U13](#), [U28](#),
[U34](#), 31, 48, 49, 82, 85, 89, 246, 248, 249, 250
- jurabib* (package) 91
`\knbccode` 2617, 2618, 2634, 2637, 2643
`\knbscode` 2539, 2540, 2563, 2567, 2573
Markus Kohm (contributor) [U31](#), 253
`KOMA` (class) [U25](#), 34, 35, 36, 254, 255
Oliver Kopp (contributor) [U31](#), 253
Scott Kostyshak (contributor) [U31](#)
Reinhard Kotucha (contributor) [U31](#), 252
Marcel Krüger (contributor) [U31](#), 51
Adam Kucharczyk (contributor) [U31](#), 246
... [U8](#), [U17](#), [U23](#), [U24](#), [U33](#), 21, 65, 120, 248
`letterspace` (package) [U1](#), [U24](#), [U33](#), [U34](#), 1, 4, 6,
12, 29, 66, 67, 69, 248, 249, 250, 252, 253, 254
`\letterspacefont` 2724
`lineno` (package) 28
`listings` (package) 31, 36, 49, 78, 86, 247, 248, 251, 255
`lmodern` (package) [U7](#), 155
`\LoadMicrotypeFile` .. [U20](#), [U34](#), 44, 80, [4372](#), [5521](#)
`\lrcode` 214, 266, 1689,
1729, 1732, 1947, 1950, 1984, 1985, 1996, 2751
`\lslig` .. [U16](#), [U23](#), [U24](#), [U33](#), 59, 66, 74, 2885, 2888
`\lssstyle` [U23](#),

- [U33](#), [56](#), [69](#), [74](#), [248](#), [1187](#), [1215](#), [2233](#),
[2866](#), [3075](#), [3081](#), [3086](#), [3089](#), [3099](#), [3103](#), [5524](#)
ltxdoc (class) [56](#), [57](#), [255](#)
luainputenc (package) [32](#), [254](#)
luaotfload (package) [U28](#),
[U32](#), [18](#), [19](#), [25](#), [60](#), [65](#), [76](#), [84](#), [251](#), [252](#), [253](#)
LuaTeX (engine) [U1](#), [U3](#), [U6](#),
[U7](#), [U8](#), [U11](#), [U12](#), [U13](#), [U14](#), [U15](#), [U16](#), [U19](#),
[U23](#), [U24](#), [U25](#), [U26](#), [U28](#), [U29](#), [U30](#), [U32](#),
[U33](#), [1](#), [4](#), [7](#), [13](#), [14](#), [15](#), [16](#), [18](#), [19](#), [21](#), [25](#),
[29](#), [37](#), [38](#), [39](#), [51](#), [60](#), [64](#), [66](#), [73](#), [75](#), [76](#), [82](#),
[108](#), [119](#), [126](#), [128](#), [129](#), [133](#), [135](#), [138](#), [141](#),
[145](#), [154](#), [223](#), [249](#), [250](#), [251](#), [252](#), [253](#), [254](#), [256](#)
luatexbase (package) [18](#), [251](#), [252](#), [253](#)
luatexja (package) [U28](#), [91](#), [252](#)
luatextra (package) [251](#)
- M** *Artur A. Marczuk* (contributor) [U31](#), [256](#)
Ronnie Marksch (contributor) [U31](#), [252](#)
marvosym (package) [U21](#), [U34](#), [4](#), [146](#), [196](#), [248](#)
Sam Mason (contributor) [U31](#), [252](#)
mathastext (package) [30](#), [251](#)
mathdesign (package) [U21](#), [140](#), [247](#)
mathtools (package) [U26](#), [35](#), [255](#)
Max (contributor) [U31](#), [252](#)
Meho R (contributor) [U31](#), [251](#)
Aman Mehra (contributor) [U31](#), [253](#), [254](#)
Peter Meier (contributor) [U31](#), [250](#)
memoir (class) [U22](#), [U25](#), [34](#), [36](#), [86](#), [91](#)
Michalis Miatidis (contributor) [U31](#), [248](#)
microtype-show (package) [U26](#), [U32](#), [4](#), [8](#), [254](#)
MicrotypeHook [U20](#), [U34](#), [123](#), [5326](#)
\microtypecontext [U21](#), [U33](#), [U34](#), [44](#), [48](#), [95](#),
[249](#), [1185](#), [2237](#), [3984](#), [4003](#), [5523](#), [5880](#), [5883](#)
microtypecontext (environment) [U22](#)
\microtypesetup [U4](#), [U8](#),
[U34](#), [27](#), [47](#), [124](#), [1412](#), [1413](#), [2237](#), [5331](#), [5522](#)
MiKTeX (distribution) [U30](#)
Daniel Benjamin Miller (contributor) [U30](#), [253](#), [254](#)
miniltx (package) [U24](#), [U33](#), [4](#), [12](#), [28](#), [250](#)
minimal (class) [27](#), [37](#), [252](#)
MinionPro (package) [98](#)
MisterFiLou (contributor) [U31](#), [254](#)
Frank Mittelbach (contributor) [U31](#), [30](#), [253](#), [254](#)
mmodern (package) [U21](#), [138](#), [253](#)
MT@font [326](#), [654](#), [1155](#),
[1442](#), [1465](#), [1505](#), [1507](#), [1649](#), [1807](#), [1821](#),
[2687](#), [2690](#), [2691](#), [2693](#), [2696](#), [3357](#), [3429](#), [3896](#)
MT@abbr@ex [1010](#)
MT@abbr@ex@c [1010](#)
MT@abbr@ex@inh [1010](#)
MT@abbr@kn [1010](#)
MT@abbr@kn@c [1010](#)
MT@abbr@kn@inh [1010](#)
MT@abbr@nl [1010](#)
MT@abbr@pr [1010](#)
MT@abbr@pr@c [1010](#)
MT@abbr@pr@inh [1010](#)
MT@abbr@sp [1010](#)
MT@abbr@sp@c [1010](#)
MT@abbr@sp@inh [1010](#)
MT@abbr@tr [1010](#)
MT@abbr@tr@c [1010](#)
MT@active@features
[134](#), [1476](#), [3933](#), [3943](#), [3955](#), [3968](#), [4012](#),
[4022](#), [4404](#), [5554](#), [5682](#), [5752](#), [5784](#), [5872](#), [5873](#)
MT@addto@annot [105](#)
MT@addto@setup [1055](#), [1056](#), [1139](#), [1317](#), [1924](#),
[2253](#), [3666](#), [3839](#), [3985](#), [3986](#), [4632](#), [4650](#),
[5142](#), [5250](#), [5332](#), [5515](#), [5829](#), [5871](#), [5896](#), [5938](#)
MT@adjustspacing [5392](#), [5393](#), [5683](#)
MT@afteraftergroup [29](#), [2780](#), [2788](#), [2797](#)
- \MT@append@patch*
. [1272](#), [1319](#), [1322](#), [1325](#), [1329](#), [1331](#),
[1332](#), [1333](#), [1334](#), [1335](#), [1336](#), [1337](#), [1338](#),
[1339](#), [1340](#), [1341](#), [1342](#), [1343](#), [1348](#), [1412](#), [1413](#)
\MT@apply@patch [1195](#), [1286](#), [5262](#), [5471](#)
\MT@auto [2413](#), [5622](#), [5637](#), [5647](#), [5650](#), [5654](#)
\MT@auto@ [2413](#), [2436](#), [2481](#)
\MT@autofalse [542](#), [5635](#), [5646](#)
\MT@autotru [542](#), [5285](#), [5288](#), [5662](#)
\MT@babelfalse [550](#)
\MT@babeltrue [550](#)
\MT@begin@catcodes
. [3298](#), [3299](#), [3336](#), [3984](#), [3988](#), [4001](#),
[4054](#), [4279](#), [4310](#), [4341](#), [4355](#), [4380](#), [4399](#),
[4431](#), [4447](#), [4469](#), [4490](#), [4504](#), [4767](#), [5301](#), [5302](#)
\MT@car [2116](#), [2123](#)
\MT@cat [179](#), [195](#), [1920](#), [3830](#), [3832](#)
\MT@cfg@catcodes [1202](#), [1942](#), [3321](#), [3338](#)
\MT@cfg@prefix [3302](#), [3303](#),
[3309](#), [3310](#), [3314](#), [4377](#), [4381](#), [4382](#), [4385](#), [4391](#)
\MT@char [45](#), [81](#),
[245](#), [1714](#), [1715](#), [1729](#), [1732](#), [1739](#), [1742](#),
[1744](#), [1750](#), [1752](#), [1770](#), [1771](#), [1774](#), [1947](#),
[1948](#), [1949](#), [1950](#), [1951](#), [2467](#), [2468](#), [2470](#),
[2471](#), [2509](#), [2510](#), [2511](#), [2539](#), [2540](#), [2546](#),
[2547](#), [2553](#), [2554](#), [2557](#), [2558](#), [2563](#), [2564](#),
[2565](#), [2566](#), [2567](#), [2568](#), [2617](#), [2618](#), [2624](#),
[2625](#), [2628](#), [2629](#), [2634](#), [2635](#), [2636](#), [2637](#),
[2638](#), [3195](#), [3196](#), [3198](#), [3458](#), [3474](#), [3482](#),
[3484](#), [3494](#), [3496](#), [3504](#), [3505](#), [3510](#), [3514](#),
[3515](#), [3520](#), [3522](#), [3528](#), [3530](#), [3533](#), [3537](#),
[3541](#), [3545](#), [3703](#), [3706](#), [3708](#), [3710](#), [3712](#),
[3775](#), [3779](#), [3784](#), [3786](#), [3789](#), [3791](#), [3797](#),
[4820](#), [4821](#), [4822](#), [4827](#), [4828](#), [4830](#), [4833](#), [4863](#)
\MT@char@ [45](#), [83](#), [1776](#), [1777](#), [1780](#), [1816](#),
[1817](#), [3458](#), [3465](#), [3469](#), [3474](#), [3523](#), [3597](#),
[3599](#), [3605](#), [3606](#), [3608](#), [3621](#), [3622](#), [3625](#),
[3626](#), [3629](#), [3630](#), [3634](#), [3636](#), [3709](#), [3711](#),
[3738](#), [3741](#), [3745](#), [3751](#), [3758](#), [3760](#), [3787](#), [3804](#)
\MT@charstring [3471](#), [3708](#), [3723](#), [3750](#)
\MT@charxstring [3753](#), [3791](#)
\MT@check@active@set
. [5312](#), [5560](#), [5700](#), [5741](#), [5761](#), [5788](#)
\MT@check@font [1438](#), [3923](#), [4014](#)
\MT@check@font@cx [3940](#), [4014](#)
\MT@check@MT@version [832](#), [5266](#)
\MT@check@range [4988](#), [4990](#)
\MT@check@range@ [4990](#), [4991](#)
\MT@check@rlist [4937](#), [4981](#)
\MT@check@rlist@ [4981](#), [4982](#)
\MT@check@step [5688](#)
\MT@checklist@ [1543](#), [1562](#), [3180](#)
\MT@checklist@family [1580](#)
\MT@checklist@font [1624](#)
\MT@checklist@size [1608](#)
\MT@checksetup [5343](#), [5354](#), [5361](#), [5380](#), [5416](#)

- \MT@clear@options 391, 529, 1045, 5508
 \MT@clist@break 851, 1575, 1603, 1619, 1633
 \MT@clist@function 851
 \MT@cnt@encoding 4876, 4884, 4885
 \MT@cnt@family 4882, 4891, 4892
 \MT@cnt@series 4889, 4898, 4899
 \MT@cnt@shape 4896, 4904, 4905
 \MT@config@file 5290, 5299, 5300, 5304, 5305, 5308
 \MT@context 3379, 3391, 3410, 3437
 \MT@copy@font 1425,
 1461, 4628, 4634, 4646, 4653, 5151, 5153, 5544
 \MT@copy@font@ . 1461, 4628, 4634, 4646, 4653, 5151
 \MT@count 45, 245, 762, 763,
 994, 1762, 1770, 1772, 1774, 1778, 1780,
 1783, 1789, 1794, 1795, 1799, 1811, 1843, 1905
 \MT@csq@eqgroup 1169, 2063
 \MT@curr@file 3303, 3304, 3310, 3311,
 4382, 4383, 4602, 4604, 4607, 4791, 5304, 5304, 5490
 \MT@curr@list@name 1809, 1846, 1936, 2477, 3192,
 3277, 3283, 3448, 3799, 3807, 3812, 3820, 3826
 \MT@curr@ls 2759, 2889, 2895
 \MT@curr@set@name 4078, 4080, 4081,
 4082, 4084, 4087, 4088, 4093, 4098, 4102,
 4103, 4112, 4137, 4141, 4170, 4198, 4221,
 4227, 4231, 4232, 4484, 4560, 4581, 4596,
 4615, 4685, 4689, 4696, 4699, 4701, 4707,
 4710, 4713, 4725, 4739, 4748, 4756, 4792, 4794
 \MT@declare@char@inh 4773, 4781, 4789
 \MT@declare@sets 4061, 4066, 4079, 4406
 \MT@DeclareMicrotypeAlias 4356, 4358
 \MT@DeclareMicrotypeSetDefault 4311, 4313
 \MT@DeclareSet 4057, 4059, 4075
 \MT@DeclareSetAndUseIt 4056, 4074
 \MT@DeclareVariants 4343, 4344, 4346
 \MT@def@bool@opt 5105,
 5118, 5119, 5123, 5137, 5149, 5159, 5171
 \MT@def@n 641, 4049, 4050
 \MT@default@ex@set 4328
 \MT@default@kn@set 4328
 \MT@default@pr@set 4328
 \MT@default@sp@set 4328
 \MT@default@tr@set 4328
 \MT@define@code@key
 4528, 4588, 4590, 4591, 4801, 4803, 4804
 \MT@define@code@key@family 4539, 4589, 4802
 \MT@define@code@key@font 4565, 4593, 4806
 \MT@define@code@key@size 4553, 4592, 4805
 \MT@define@opt@key 4594,
 4617, 4618, 4619, 4620, 4752, 4753, 4754
 \MT@define@optionX 5331, 5389, 5392
 \MT@define@optionX@ 5406, 5435, 5437, 5438
 \MT@define@patch 1249,
 1270, 1318, 1328, 1347, 1355, 1379, 1411
 \MT@define@set@key@ . 4091, 4271, 4272, 4273, 4274
 \MT@define@set@key@font 4219, 4276
 \MT@define@set@key@size 4130, 4275
 \MT@detokenize@c 686, 3466, 3704, 3719
 \MT@detokenize@n 686, 3781
 \MT@dimen@six 1497,
 1763, 1799, 1843, 1982, 1991, 2974, 2976, 3145
 \MT@dinfo 93
 \MT@dinfo@list
 1560, 1570, 1573, 1578, 1587, 1590, 1596,
 1598, 1606, 1614, 1617, 1622, 1632, 1635, 1639
 \MT@dinfo@nl 93
 \MT@disablefalse 546
 \MT@DisableLigatures 4396
 \MT@disabletrue 546, 5174, 5175
 \MT@do@font 233, 240, 253, 299,
 316, 330, 961, 1691, 2441, 2576, 2645, 2751
 \MT@documentfalse 582
 \MT@documenttrue 582, 5875
 \MT@dofalse ... 1535, 1547, 1574, 1591, 1618, 1636
 \MT@dofont@function 965, 976
 \MT@dotrue 1535, 1538, 1571, 1588, 1597, 1615, 3175
 \MT@draftfalse 545
 \MT@drafttrue 545
 \MT@edef@n 643, 3415, 3441, 4039,
 4534, 4548, 4604, 4607, 4610, 4790, 4968, 5209
 \MT@encoding
 246, 1512, 1628, 1653, 1655, 1658, 3374,
 3386, 3466, 3704, 3719, 3780, 3811, 3818, 3826
 \MT@end@catcodes 3300, 3340, 3985, 3992,
 4002, 4072, 4295, 4326, 4352, 4370, 4388,
 4408, 4442, 4466, 4485, 4501, 4515, 4784, 5303
 \MT@engine 402,
 516, 522, 526, 3076, 3080, 5275, 5277, 5278
 \MT@engine@minversion 402, 523
 \MT@engine@unfitfalse 403, 413, 425, 447
 \MT@engine@unfittrue 403, 404, 416, 428
 \MT@error 73, 82, 1037, 1142, 3076,
 3087, 3266, 3286, 4168, 4304, 4333, 4414,
 4637, 4663, 4670, 5161, 5326, 5366, 5384,
 5424, 5628, 5640, 5660, 5729, 5799, 5812, 5892
 \MT@error@doesnt@work 5795
 \MT@ex@cc@name 2395,
 2397, 2483, 2484, 2495, 2501, 2502, 2514, 4450
 \MT@ex@context 1465, 4006, 4048
 \MT@ex@doc@contexts 4048
 \MT@ex@factor 551, 2414, 2487
 \MT@ex@factor@ . 2414, 2443, 2448, 2457, 2458, 2481
 \MT@ex@inh@name 2469, 2470, 2471
 \MT@ex@level 551, 5683, 5685
 \MT@ex@max 563, 2460, 2461
 \MT@ex@min 563, 2463, 2464
 \MT@ex@set@name 4297
 \MT@ex@split@val 2455
 \MT@exp@cs 305, 335, 638, 641, 644,
 647, 652, 1283, 1366, 1402, 1404, 1477,
 1479, 1488, 1612, 1751, 1830, 1833, 1863,
 1867, 1877, 1880, 1883, 1956, 2471, 2558,
 2629, 2691, 2800, 2801, 2802, 3402, 3719,
 3959, 3969, 3970, 3974, 4025, 4036, 4110,
 4136, 4559, 4579, 4830, 4832, 4939, 4986, 5703
 \MT@exp@gcs 638, 642, 646, 648, 653
 \MT@exp@one@n ... 302, 318, 655, 691, 858, 1429,
 1431, 1489, 1566, 1629, 2132, 2134, 2195,
 2210, 2323, 2324, 2325, 2339, 2342, 2345,
 2347, 2367, 2378, 2383, 2672, 3307, 3923,
 3926, 3944, 3956, 3962, 3980, 4022, 4033,
 4048, 4066, 4270, 4289, 4320, 4375, 4781, 5880
 \MT@exp@two@c 657, 675, 680, 692, 893, 1428, 1469,
 1471, 1473, 1474, 1483, 2116, 2131, 2865,
 2904, 2910, 3463, 3470, 3707, 3774, 3789, 3790
 \MT@exp@two@n 659, 1584, 1593, 4097, 4226
 \MT@expandfont 2399, 2417, 2421
 \MT@expansion 1445, 2386, 5720
 \MT@expansionfalse 541, 5578, 5583, 5677
 \MT@expansiontrue 541, 5284

- \MT@extra@context [4048](#), [4434](#), [4450](#), [4473](#),
 [4493](#), [4507](#), [4574](#), [4577](#), [4578](#), [4580](#), [4621](#),
 [4629](#), [4635](#), [4647](#), [4652](#), [4763](#), [4936](#), [4940](#),
 [4943](#), [4946](#), [4947](#), [4952](#), [4957](#), [4958](#), [4960](#), [4987](#)
 \MT@extra@inputenc [4761](#), [4788](#), [4793](#), [4794](#)
 \MT@factor [120](#)
 \MT@factor@default [575](#), [5216](#), [5557](#)
 \MT@family [1429](#), [1512](#),
 [1585](#), [1628](#), [1653](#), [1655](#), [1658](#), [3375](#), [4366](#), [4367](#)
 \MT@familyalias [1430](#),
 [1431](#), [1524](#), [1592](#), [1594](#), [3385](#), [3387](#), [4368](#)
 \MT@feat . [41](#), [173](#), [1535](#), [1717](#), [1825](#), [1827](#), [1829](#),
 [1830](#), [1832](#), [1833](#), [1839](#), [1841](#), [1844](#), [1852](#),
 [1853](#), [1854](#), [1855](#), [1856](#), [1858](#), [1860](#), [1861](#),
 [1862](#), [1863](#), [1864](#), [1867](#), [1868](#), [1873](#), [1877](#),
 [1880](#), [1883](#), [1886](#), [1887](#), [1888](#), [1897](#), [1898](#),
 [1903](#), [1921](#), [2870](#), [3264](#), [3266](#), [3271](#), [3273](#),
 [3278](#), [3280](#), [3286](#), [3411](#), [3413](#), [3415](#), [3419](#),
 [3423](#), [3427](#), [3428](#), [3431](#), [3439](#), [3441](#), [3442](#),
 [3444](#), [3451](#), [3455](#), [3800](#), [3801](#), [3830](#), [3832](#), [4851](#)
 \MT@features [1030](#),
 [4048](#), [4061](#), [4270](#), [4284](#), [4315](#), [4599](#), [4772](#)
 \MT@features@long [1030](#), [1033](#), [1038](#), [4019](#), [4786](#)
 \MT@file@list
 [3293](#), [3296](#), [3301](#), [3307](#), [3312](#), [4375](#), [4379](#)
 \MT@find@file [1429](#), [1431](#), [3293](#)
 \MT@fix@catcode [5](#), [1296](#), [1297](#)
 \MT@fix@font@set [4120](#), [5528](#)
 \MT@fix@fontdimen@six [1497](#), [1523](#)
 \MT@font [37](#),
 [42](#), [65](#), [92](#), [249](#), [279](#), [654](#), [988](#), [990](#), [1422](#),
 [1428](#), [1483](#), [1490](#), [1499](#), [1500](#), [1501](#), [1504](#),
 [1506](#), [1509](#), [1689](#), [1690](#), [1729](#), [1732](#), [1739](#),
 [1742](#), [1770](#), [1771](#), [1774](#), [1777](#), [1780](#), [1806](#),
 [1811](#), [1817](#), [1911](#), [1914](#), [1947](#), [1948](#), [1950](#),
 [1951](#), [1980](#), [1981](#), [1984](#), [1985](#), [1989](#), [1990](#),
 [1993](#), [1994](#), [1996](#), [1997](#), [2426](#), [2430](#), [2436](#),
 [2441](#), [2467](#), [2468](#), [2509](#), [2511](#), [2539](#), [2540](#),
 [2546](#), [2547](#), [2553](#), [2554](#), [2563](#), [2564](#), [2565](#),
 [2567](#), [2568](#), [2573](#), [2574](#), [2575](#), [2617](#), [2618](#),
 [2624](#), [2625](#), [2634](#), [2635](#), [2637](#), [2638](#), [2643](#),
 [2644](#), [2672](#), [2676](#), [2757](#), [2758](#), [3184](#), [3481](#),
 [3536](#), [3834](#), [3891](#), [3892](#), [3893](#), [3895](#), [3910](#),
 [3911](#), [3923](#), [3924](#), [3929](#), [3944](#), [3962](#), [3972](#), [3981](#)
 \MT@font@copy
 [1465](#), [1473](#), [1475](#), [1483](#), [1484](#), [1485](#), [1495](#)
 \MT@font@list
 [3834](#), [3923](#), [3924](#), [3926](#), [3928](#), [3929](#), [4013](#)
 \MT@font@orig [1467](#)
 \MT@font@sets [4101](#), [4120](#), [4230](#), [5528](#)
 \MT@fontspecfalse [1107](#)
 \MT@fontspectrue [1107](#), [1108](#), [1113](#), [1114](#), [1152](#)
 \MT@gdef@n [641](#), [1255](#), [1256](#), [1263](#), [1264](#), [4427](#),
 [4441](#), [4465](#), [4500](#), [4514](#), [4701](#), [4725](#), [4739](#),
 [4796](#), [4869](#), [4870](#), [4871](#), [4872](#), [4873](#), [4874](#), [5320](#)
 \MT@get@axis [4244](#), [4245](#), [4246](#), [4247](#), [4257](#)
 \MT@get@basefamily [3306](#), [3341](#)
 \MT@get@basefamily@ [3347](#), [3350](#)
 \MT@get@char@unit [44](#), [1716](#), [1875](#), [1901](#), [2391](#)
 \MT@get@charwd [83](#), [1768](#), [1799](#), [1878](#), [1901](#)
 \MT@get@ex@opt [47](#), [2390](#), [2408](#), [2481](#)
 \MT@get@ex@opt@ [2489](#), [2490](#), [2491](#), [2493](#), [2500](#)
 \MT@get@font [4225](#), [4235](#)
 \MT@get@font@ [4236](#), [4241](#), [4586](#)
 \MT@get@font@and@size [100](#), [4570](#), [4585](#)
 \MT@get@font@dimen [1804](#), [1881](#)
 \MT@get@highlevel [4096](#), [4106](#), [4259](#), [4533](#), [4544](#)
 \MT@get@inh@list [325](#), [1665](#), [2393](#), [2526](#), [2604](#), [3437](#)
 \MT@get@listname [3356](#), [3413](#), [3439](#)
 \MT@get@listname@ [3356](#)
 \MT@get@ls@basefont [2715](#), [2894](#), [2901](#)
 \MT@get@MT@version [5266](#)
 \MT@get@opt [1663](#), [1850](#), [2524](#), [2602](#)
 \MT@get@prot [2048](#), [2053](#), [2061](#), [2063](#), [2174](#), [2175](#), [2191](#)
 \MT@get@range [4134](#), [4144](#), [4557](#)
 \MT@get@size [4151](#), [4156](#), [4165](#), [4184](#), [4255](#)
 \MT@get@size@ [4191](#), [4202](#)
 \MT@get@size@@ [4202](#)
 \MT@get@slot [31](#), [37](#), [1713](#), [3194](#), [3458](#), [4819](#), [4826](#)
 \MT@get@slot@ [3475](#), [3479](#)
 \MT@get@space@unit
 [62](#), [1875](#), [2025](#), [2537](#), [2544](#), [2551](#), [2615](#), [2622](#)
 \MT@get@tr@opt [47](#), [2702](#), [2833](#)
 \MT@get@tr@opt@ [2853](#), [2854](#), [2855](#), [2856](#), [2858](#)
 \MT@get@unit [1883](#), [1891](#), [2842](#)
 \MT@get@unit@ [1891](#)
 \MT@getkey [5495](#), [5510](#)
 \MT@getthird [2124](#), [2130](#)
 \MT@glet [635](#), [648](#),
 [1153](#), [1469](#), [2801](#), [2904](#), [2935](#), [3081](#), [3089](#),
 [3889](#), [3995](#), [4014](#), [4015](#), [4016](#), [4017](#), [4368](#),
 [4628](#), [4646](#), [4879](#), [5151](#), [5153](#), [5843](#), [5849](#), [5944](#)
 \MT@glet@nc
 [647](#), [1507](#), [1658](#), [2696](#), [2948](#), [3136](#), [3451](#),
 [3911](#), [4013](#), [4084](#), [4087](#), [4093](#), [4221](#), [4458](#),
 [4685](#), [4689](#), [4696](#), [4699](#), [4707](#), [4710](#), [4713](#), [4748](#)
 \MT@glet@nn [652](#), [4032](#), [5317](#)
 \MT@gobble@to@nil [2051](#), [2324](#)
 \MT@has@inh@prefix [4818](#), [4844](#)
 \MT@if@expanding@F [683](#), [3885](#)
 \MT@if@expanding@F@ [683](#)
 \MT@if@false [539](#), [1124](#), [1131](#),
 [1156](#), [1177](#), [1381](#), [3682](#), [3949](#), [4985](#), [5911](#), [5918](#)
 \MT@if@list@exists
 [59](#), [1650](#), [2389](#), [2407](#), [2523](#), [2601](#), [2701](#), [3410](#)
 \MT@if@luaotf@font
 [942](#), [964](#), [2717](#), [2749](#), [2758](#), [3197](#), [3210](#)
 \MT@if@outer@next [3015](#),
 [3030](#), [3034](#), [3037](#), [3042](#), [3047](#), [3048](#), [3049](#), [3050](#)
 \MT@if@true [539](#), [1125](#), [1126](#), [1127](#), [1132](#),
 [1133](#), [1134](#), [1157](#), [1158](#), [1159](#), [1190](#), [1197](#),
 [1199](#), [1385](#), [3683](#), [3684](#), [3942](#), [4994](#), [4998](#),
 [5006](#), [5011](#), [5912](#), [5913](#), [5914](#), [5915](#), [5916](#), [5919](#)
 \MT@ifdefined@c@T
 [296](#), [663](#), [1128](#), [1162](#), [1673](#), [1749](#),
 [1788](#), [1954](#), [2469](#), [2556](#), [2627](#), [2743](#), [2847](#),
 [3652](#), [4122](#), [4366](#), [4793](#), [4926](#), [5326](#), [5330](#), [5947](#)
 \MT@ifdefined@c@TF [327](#), [663](#), [1072](#),
 [1384](#), [1406](#), [2035](#), [2705](#), [2727](#), [2730](#), [2741](#),
 [2889](#), [2971](#), [3142](#), [3189](#), [3190](#), [3414](#), [3440](#), [4935](#)
 \MT@ifdefined@n@T
 [301](#), [331](#), [340](#), [663](#), [1061](#), [1563](#), [1581](#),
 [1609](#), [1625](#), [1674](#), [1750](#), [1886](#), [1922](#), [1955](#),
 [2470](#), [2495](#), [2557](#), [2628](#), [2837](#), [2845](#), [2859](#),
 [3393](#), [3669](#), [3687](#), [3829](#), [4081](#), [4361](#), [4453](#),
 [4611](#), [4885](#), [4892](#), [4899](#), [4905](#), [4936](#), [4946](#), [5495](#)
 \MT@ifdefined@n@TF
 [663](#), [1251](#), [1262](#), [1281](#), [1289](#), [1365](#), [1524](#),
 [1541](#), [1564](#), [1582](#), [1610](#), [1626](#), [1653](#), [1852](#),
 [1860](#), [2483](#), [2501](#), [2690](#), [2799](#), [3178](#), [3271](#),

- 3381, 3466, 4109, 4298, 4301, 4329, 4602,
 4912, 4967, 4976, 5046, 5313, 5316, 5701, 5878
 \MT@ifdim 785,
 913, 914, 918, 919, 4167, 4176, 4992, 4993,
 4994, 4996, 4997, 5004, 5005, 5006, 5009, 5010
 \MT@ifdimen 747, 4193, 4688, 4698, 4712, 5225
 \MT@ifempty 260,
 698, 1689, 1690, 1725, 1735, 1746, 1747,
 2014, 2015, 2315, 2323, 2324, 2536, 2543,
 2550, 2573, 2574, 2575, 2587, 2588, 2589,
 2591, 2592, 2593, 2614, 2621, 2643, 2644,
 2660, 2661, 2954, 2955, 2963, 2979, 3111,
 3134, 3135, 3147, 3717, 4024, 4060, 4064,
 4108, 4118, 4145, 4146, 4159, 4160, 4249,
 4260, 4283, 4287, 4314, 4318, 4403, 4480,
 4595, 4601, 4621, 4627, 4635, 4645, 4652,
 4723, 4771, 4778, 5059, 5090, 5293, 5340, 5413
 \MT@ifint 711, 3628, 4481, 4724, 5208, 5218
 \MT@ifstreq 794, 1553, 1928,
 2099, 2101, 2493, 3265, 3412, 3427, 3739,
 3742, 3750, 3759, 3979, 4224, 4569, 4687,
 4709, 4711, 4773, 4780, 4851, 4852, 4853,
 4854, 4927, 4947, 5018, 5062, 5064, 5067,
 5070, 5092, 5094, 5108, 5109, 5174, 5175,
 5176, 5185, 5186, 5191, 5195, 5224, 5236,
 5238, 5244, 5246, 5270, 5342, 5349, 5353,
 5360, 5415, 5421, 5465, 5467, 5477, 5479, 5771
 \MT@in@clist 302, 318, 879, 1033, 1062,
 1290, 1307, 1566, 1584, 1593, 1629, 2672,
 3296, 3307, 3923, 3926, 3944, 3956, 4022, 4375
 \MT@in@rlist 907, 1612, 3402
 \MT@in@rlist@ 907
 \MT@in@rlist@@ 907
 \MT@in@tlist 895, 3353, 4033
 \MT@in@tlist@ 895
 \MT@inannotfalse 106
 \MT@inannottrue 106
 \MT@increment 994, 4884, 4891, 4898, 4904
 \MT@info 82, 97, 102, 1155,
 1195, 1196, 1245, 1248, 1505, 5192, 5456, 5460
 \MT@info@missing@char 1783, 1814, 3532
 \MT@info@nl 82,
 98, 99, 103, 1073, 1370, 1654, 1815, 5183,
 5193, 5300, 5314, 5318, 5540, 5544, 5556,
 5563, 5684, 5721, 5740, 5745, 5754, 5764,
 5787, 5791, 5804, 5819, 5888, 5899, 5925
 \MT@info@nottracking 1443, 1554, 1558
 \MT@info@nottracking@ 1554, 1558
 \MT@inh@do 3450, 4807
 \MT@inh@feat 4761, 4771, 4776, 4787
 \MT@inh@prefix 4829, 4839, 4844
 \MT@inh@prefix@ 4833, 4852, 4853, 4854
 \MT@inh@split 4809, 4813
 \MT@inlist@false 879, 883, 896, 908, 3952
 \MT@inlist@true 879, 885, 903, 915, 920, 3952
 \MT@is@active 3463, 3641
 \MT@is@active@hook 3654, 3663
 \MT@is@char 89, 3470, 3707, 3723
 \MT@is@charx 3732, 3753, 3790
 \MT@is@composite 3468, 3777
 \MT@is@feature 1032, 4065, 4288, 4319
 \MT@is@letter 3464, 3585, 3712, 3786
 \MT@is@number 3611, 3616, 3774
 \MT@is@opt@char 3706, 3716
 \MT@is@symbol 3467, 3702
 \MT@is@tlig 3710, 3767
 \MT@is@uni@comp 3784, 3796
 \MT@is@xchar 3743, 3749
 \MT@iterate 929
 \MT@kerning 1449, 2598, 5790
 \MT@kerningfalse 548
 \MT@kerningtrue 548
 \MT@kn@c@name 2606, 2608, 2651, 4507
 \MT@kn@context 4007, 4048
 \MT@kn@doc@contexts 4048
 \MT@kn@factor 551
 \MT@kn@factor@ 1852
 \MT@kn@inh@name 2627, 2628, 2629
 \MT@kn@max 563
 \MT@kn@min 563
 \MT@kn@setname 4297
 \MT@kn@split@val 2612
 \MT@kn@unit 557
 \MT@kn@unit@ 1860, 2654
 \MT@led@unhbox@line 1073
 \MT@ledmac@setup 1067, 1148, 1149, 1150
 \MT@let@cn 649,
 652, 653, 1284, 1525, 1669, 2397, 2484,
 2530, 2608, 2838, 2846, 3264, 3280, 3407,
 3411, 3444, 3675, 3693, 3891, 4188, 4615, 4792
 \MT@let@nc 647,
 1239, 1282, 1888, 3056, 3423, 3455, 3947,
 4969, 4974, 5068, 5071, 5356, 5363, 5802, 5817
 \MT@let@nn 652, 1853, 1858, 1861, 1873, 2502, 2505, 2860
 \MT@letterspace 578,
 2705, 2846, 2848, 2850, 5852, 5853, 5855
 \MT@letterspace@ 1700, 1788, 1789,
 2705, 2706, 2709, 2712, 2724, 2729, 2737,
 2750, 2751, 2752, 2759, 2768, 2810, 2976,
 2979, 3097, 3112, 3113, 3114, 3115, 3148, 3151
 \MT@letterspace@default 578, 5853
 \MT@listname 327, 329, 331, 336, 340, 3356, 3404,
 3407, 3414, 3415, 3417, 3419, 3440, 3441,
 3443, 3444, 3448, 3451, 4830, 4832, 4839, 4856
 \MT@listname@count 4598, 4603, 4605
 \MT@load@inputenc 1929, 1932, 1941
 \MT@load@list .. 185, 1667, 2395, 2528, 2606, 3260
 \MT@loop 929, 937, 987, 4211, 4973
 \MT@lower 4138, 4144, 4560
 \MT@ls@adjust 67, 2766, 3108
 \MT@ls@adjust@ 3092, 3116
 \MT@ls@adjust@empty 3094, 3108
 \MT@ls@adjust@relax 3095, 3108
 \MT@ls@basefont 2901, 2910, 2913, 2914
 \MT@ls@fontspec@font 2720, 2808
 \MT@ls@outer@k 2769,
 2774, 2782, 2992, 3006, 3053, 3071, 3156
 \MT@ls@set@ls 3099, 3102, 3108
 \MT@ls@too@large 3115, 3118, 5855
 \MT@ls@font 2711,
 2720, 2724, 2737, 2738, 2747, 2751, 2752,
 2757, 2865, 2876, 2924, 2926, 2941, 2942,
 2949, 2950, 2957, 2959, 2965, 2980, 3149, 3152
 \MT@ltx@pickupfont
 183, 199, 1183, 1228, 1233, 3906, 3914
 \MT@lua 162, 430, 493, 593, 602, 735,
 772, 820, 833, 944, 966, 2810, 3198, 3211, 3486
 \MT@lua@copy@font 1494
 \MT@lua@copyfont 1474, 1494

- \MT@luatex@no [491](#), [591](#)
 \MT@map@clist@ [851](#)
 \MT@map@clist@c ... [851](#), [1476](#), [3193](#), [3943](#), [3955](#),
 [3968](#), [4012](#), [4019](#), [4061](#), [4284](#), [4315](#), [4599](#),
 [4772](#), [4776](#), [4786](#), [5258](#), [5262](#), [5471](#), [5483](#), [5493](#)
 \MT@map@clist@n [851](#),
 [1540](#), [3177](#), [4048](#), [4063](#), [4083](#), [4094](#), [4132](#),
 [4222](#), [4270](#), [4286](#), [4317](#), [4347](#), [4425](#), [4531](#),
 [4542](#), [4555](#), [4567](#), [4705](#), [4721](#), [4823](#), [5054](#),
 [5057](#), [5086](#), [5088](#), [5117](#), [5203](#), [5338](#), [5411](#)
 \MT@map@tlist@ [869](#)
 \MT@map@tlist@c [305](#),
 [335](#), [869](#), [898](#), [909](#), [1479](#), [1751](#), [1956](#), [2336](#),
 [2471](#), [2558](#), [2629](#), [3360](#), [3974](#), [4937](#), [4986](#), [5528](#)
 \MT@map@tlist@n [869](#), [3332](#), [3725](#), [4878](#)
 \MT@max@char [3585](#), [3608](#)
 \MT@max@slot [3585](#), [3634](#)
 \MT@maybe@do .. [1535](#), [1642](#), [2387](#), [2521](#), [2599](#), [2674](#)
 \MT@maybe@gobble@with@tikz [1115](#), [2798](#), [2869](#)
 \MT@maybe@rem@from@list [3974](#), [3978](#)
 \MT@maybe@textcmd ... [2044](#), [2053](#), [2200](#), [2224](#), [2378](#)
 \MT@microtypecontext [3984](#)
 \MT@MT [2](#), [82](#), [85](#),
 [88](#), [90](#), [92](#), [97](#), [98](#), [100](#), [101](#), [520](#), [525](#), [834](#),
 [1044](#), [1143](#), [1145](#), [4637](#), [5268](#), [5272](#), [5275](#),
 [5277](#), [5278](#), [5294](#), [5456](#), [5460](#), [5541](#), [5892](#), [5893](#)
 \MT@MT@pickupfont [1230](#), [1235](#), [3906](#), [3916](#)
 \MT@next@listname [42](#), [3363](#), [3372](#)
 \MT@next@listname@ [3382](#), [3394](#), [3400](#)
 \MT@nil [2057](#), [2204](#), [2226](#), [2315](#), [2316](#)
 \MT@nl@ligatures [3184](#), [4396](#)
 \MT@nl@setname [3176](#), [4396](#)
 \MT@nofamilyfalse [1641](#), [1648](#)
 \MT@nofamilytrue [1641](#), [3373](#)
 \MT@noindent [2063](#)
 \MT@noligatures [248](#), [1453](#), [3172](#), [5864](#)
 \MT@noligatures@ [2924](#), [2926](#), [3184](#), [3187](#)
 \MT@noligaturesfalse [544](#)
 \MT@noligaturestrue [544](#), [4405](#)
 \MT@nonselectedfalse [2403](#), [2418](#)
 \MT@nonselectedtrue [2403](#), [2406](#)
 \MT@nopatches@list
 ... [5232](#), [5245](#), [5248](#), [5254](#), [5255](#), [5256](#), [5258](#)
 \MT@noresetfalse
 ... [3594](#), [3601](#), [3610](#), [3631](#), [3739](#), [3751](#), [3759](#)
 \MT@noresettrue [3461](#), [3594](#), [3619](#)
 \MT@old@cmd [72](#), [77](#), [78](#), [79](#), [80](#)
 \MT@opt@autofalse [5029](#)
 \MT@opt@autotrue [5029](#), [5119](#)
 \MT@opt@def@set [5045](#), [5073](#), [5098](#)
 \MT@opt@DVIfalse [5030](#)
 \MT@opt@DVITrue [5030](#), [5126](#), [5129](#)
 \MT@opt@expansionfalse [5028](#)
 \MT@opt@expansiontrue [5028](#)
 \MT@options [4518](#)
 \MT@optwarn@admissible ... [5031](#), [5110](#), [5177](#), [5195](#)
 \MT@optwarn@nan [5035](#), [5210](#), [5220](#)
 \MT@orig@add@accent [3912](#)
 \MT@orig@foreign@language [5905](#), [5907](#)
 \MT@orig@pickupfont [3840](#), [3908](#)
 \MT@orig@py@macron [1225](#), [1229](#), [1234](#)
 \MT@orig@select@language [5900](#), [5902](#)
 \MT@outer@kern [2768](#), [2772](#), [2774](#), [2775](#), [2779](#), [2787](#),
 [2895](#), [2896](#), [2898](#), [3053](#), [3054](#), [3071](#), [3129](#), [3168](#)
 \MT@outer@space
 [2762](#), [2763](#), [2939](#), [2988](#), [2990](#), [2991](#), [3004](#),
 [3005](#), [3022](#), [3023](#), [3038](#), [3039](#), [3043](#), [3044](#), [3069](#)
 \MT@patch@info [1244](#)
 \MT@patch@info@undo [1244](#), [1311](#)
 \MT@patch@name [1283](#), [1298](#)
 \MT@patch@okfalse [1244](#), [1274](#), [1278](#)
 \MT@patch@oktrue [1244](#), [1288](#), [1369](#)
 \MT@patch@patch [1272](#),
 [1320](#), [1323](#), [1357](#), [1358](#), [1359](#), [1360](#), [1363](#),
 [1364](#), [1366](#), [1376](#), [1377](#), [1390](#), [1391](#), [1392](#),
 [1393](#), [1399](#), [1400](#), [1402](#), [1404](#), [1407](#), [1408](#), [1409](#)
 \MT@patch@undef [1244](#), [1304](#)
 \MT@patch@warn [1244](#)
 \MT@patches@applied [1286](#), [1307](#), [1309](#)
 \MT@patches@def [1249](#), [5252](#), [5256](#), [5466](#), [5478](#)
 \MT@patches@list [5232](#),
 [5239](#), [5240](#), [5251](#), [5252](#), [5259](#), [5261](#), [5262](#)
 \MT@pdf@annot [7](#), [105](#)
 \MT@pdf@or@lua .. [409](#), [427](#), [524](#), [600](#), [714](#), [942](#), [5836](#)
 \MT@pdftex@no [14](#), [454](#), [590](#)
 \MT@permute
 [113](#), [4440](#), [4464](#), [4478](#), [4499](#), [4513](#), [4799](#), [4869](#)
 \MT@permute@ [4869](#)
 \MT@permute@@ [4869](#)
 \MT@permute@@@ [4869](#)
 \MT@permute@@@@ [4903](#), [4908](#)
 \MT@permute@@@@@ [4916](#), [4918](#)
 \MT@permute@define .. [4909](#), [4919](#), [4920](#), [4921](#), [4965](#)
 \MT@permute@reset [4878](#), [4971](#)
 \MT@permutelist [4434](#), [4450](#), [4476](#), [4493](#),
 [4507](#), [4574](#), [4575](#), [4580](#), [4797](#), [4931](#), [4932](#),
 [4936](#), [4940](#), [4943](#), [4946](#), [4947](#), [4948](#), [4950](#),
 [4951](#), [4952](#), [4960](#), [4961](#), [4987](#), [5019](#), [5021](#), [5022](#)
 \MT@pickupfont [3906](#)
 \MT@plain [352](#), [374](#)
 \MT@pr@c@name [246](#), [1662](#), [1667](#), [1669](#), [1672](#), [2005](#), [4434](#)
 \MT@pr@c@context [1465](#), [4006](#), [4048](#)
 \MT@pr@doc@contexts [4048](#)
 \MT@pr@factor [551](#), [5219](#), [5557](#), [5558](#)
 \MT@pr@factor@ [1852](#)
 \MT@pr@inh@name [296](#), [301](#), [306](#),
 [1673](#), [1674](#), [1749](#), [1750](#), [1752](#), [1954](#), [1955](#), [1957](#)
 \MT@pr@level [551](#), [5555](#), [5556](#)
 \MT@pr@max [563](#)
 \MT@pr@min [563](#)
 \MT@pr@setname [4297](#)
 \MT@pr@split@val [66](#), [184](#), [1721](#)
 \MT@pr@unit [557](#), [5226](#), [5559](#)
 \MT@pr@unit@ [1860](#), [2008](#)
 \MT@preset@aux [2010](#),
 [2012](#), [2014](#), [2015](#), [2018](#), [2656](#), [2658](#), [2660](#), [2661](#)
 \MT@preset@aux@factor
 ... [2010](#), [2018](#), [2587](#), [2588](#), [2589](#), [2656](#)
 \MT@preset@aux@space
 ... [2012](#), [2018](#), [2591](#), [2592](#), [2593](#), [2658](#)
 \MT@preset@ex [2496](#), [2513](#)
 \MT@preset@kn [2649](#)
 \MT@preset@kn@ [2649](#)
 \MT@preset@pr [2002](#)
 \MT@preset@pr@ [2002](#)
 \MT@preset@sp [2580](#)
 \MT@preset@sp@ [2580](#)
 \MT@ProcessOptionsWithKV [5487](#), [5511](#)

- \MT@prot@addgroup [2138](#), [2307](#)
 \MT@prot@addtoken@first [2327](#), [2330](#), [2359](#)
 \MT@prot@addtoken@next [2364](#), [2373](#)
 \MT@prot@check [2143](#), [2336](#)
 \MT@prot@check@ [2143](#)
 \MT@prot@check@C [2208](#)
 \MT@prot@check@C@ [2218](#), [2219](#), [2223](#)
 \MT@prot@check@cmds [2228](#), [2336](#)
 \MT@prot@check@E [2163](#)
 \MT@prot@check@e [2167](#)
 \MT@prot@check@eX [2171](#)
 \MT@prot@check@F [51](#), [2193](#)
 \MT@prot@check@I [2151](#)
 \MT@prot@check@l [2177](#)
 \MT@prot@check@l@ [2177](#)
 \MT@prot@check@lX [2184](#)
 \MT@prot@check@lX@ [2184](#)
 \MT@prot@check@O [2157](#)
 \MT@prot@check@S [2154](#)
 \MT@prot@check@T [2160](#)
 \MT@prot@check@X [2157](#)
 \MT@prot@get@first@group
 [2043](#), [2139](#), [2198](#), [2213](#), [2305](#)
 \MT@prot@get@first@group@ [2043](#), [2305](#)
 \MT@prot@get@first@group@etc [2140](#), [2198](#), [2213](#), [2313](#)
 \MT@prot@get@first@token [2141](#), [2309](#), [2318](#), [2322](#)
 \MT@prot@get@firstgroup [53](#),
 [2045](#), [2138](#), [2139](#), [2152](#), [2155](#), [2158](#), [2161](#), [2333](#)
 \MT@prot@get@firstgroup@etc [2139](#), [2204](#), [2226](#), [2316](#)
 \MT@prot@get@firsttoken [2139](#)
 \MT@prot@get@next@token [2142](#), [2363](#)
 \MT@prot@get@nexttoken [2139](#), [2361](#), [2382](#)
 \MT@prot@hook [1167](#), [2063](#)
 \MT@prot@ifcat
 [2108](#), [2306](#), [2314](#), [2326](#), [2329](#), [2365](#), [2366](#)
 \MT@prot@iffirstcmd
 [2114](#), [2126](#), [2127](#), [2338](#), [2341](#), [2344](#), [2347](#)
 \MT@prot@iflicrmd [2124](#), [2348](#)
 \MT@prot@ifmacro [2111](#), [2147](#), [2335](#), [2337](#)
 \MT@prot@ifmacro@ [2111](#), [2335](#)
 \MT@prot@ifx [2105](#), [2332](#)
 \MT@prot@l [2042](#), [2047](#), [2132](#), [2134](#), [2165](#),
 [2169](#), [2182](#), [2194](#), [2195](#), [2197](#), [2209](#), [2210](#),
 [2212](#), [2325](#), [2339](#), [2342](#), [2345](#), [2347](#), [2367](#), [2383](#)
 \MT@prot@l@ [2042](#), [2047](#)
 \MT@prot@l@etc [2051](#), [2194](#), [2197](#), [2209](#), [2212](#)
 \MT@prot@next [53](#), [2106](#), [2109](#),
 [2112](#), [2116](#), [2130](#), [2139](#), [2140](#), [2141](#), [2142](#), [2145](#)
 \MT@prot@r [2058](#)
 \MT@prot@toks [51](#), [2041](#), [2051](#), [2203](#), [2225](#), [2324](#)
 \MT@protrudechars [5389](#), [5393](#), [5555](#), [5742](#)
 \MT@protrusion [1444](#), [1642](#), [5562](#)
 \MT@protrusionfalse [540](#)
 \MT@protrusiontrue [540](#), [5279](#)
 \MT@rba@expansion [1025](#)
 \MT@rba@kerning [1025](#)
 \MT@rba@protrusion [1025](#)
 \MT@rba@spacing [1025](#)
 \MT@rba@tracking [1025](#)
 \MT@redefine@patch [1259](#), [1271](#)
 \MT@redefined@patches [1259](#), [1415](#)
 \MT@register@font [1457](#), [3924](#), [4015](#)
 \MT@register@font@cx [3967](#), [4015](#)
 \MT@register@subst@font [3897](#), [3925](#), [4016](#)
 \MT@register@subst@font@cx [3954](#), [4016](#)
 \MT@rem@from@clist
 [890](#), [1309](#), [1489](#), [3929](#), [3962](#), [3980](#), [4523](#), [5259](#)
 \MT@rem@from@list [1479](#), [1487](#)
 \MT@rem@last@space [686](#)
 \MT@remember@patch [1273](#), [1277](#), [1280](#)
 \MT@remove@tlig [3770](#), [3773](#)
 \MT@repeat [929](#), [939](#), [990](#), [4214](#), [4979](#)
 \MT@requires@latex [373](#), [392](#), [1053](#), [1179](#),
 [2864](#), [2891](#), [3838](#), [5037](#), [5202](#), [5488](#), [5492](#), [5943](#)
 \MT@requires@luatex
 [585](#), [635](#), [1435](#), [1502](#), [1698](#), [2422](#),
 [2423](#), [2445](#), [2492](#), [2668](#), [3187](#), [3935](#), [4738](#),
 [4744](#), [5122](#), [5394](#), [5408](#), [5573](#), [5650](#), [5658](#), [5737](#)
 \MT@requires@pdftex [585](#), [713](#), [748](#), [795](#), [1068](#),
 [1424](#), [1434](#), [1448](#), [1452](#), [1463](#), [1507](#), [1697](#),
 [1760](#), [1786](#), [2444](#), [2520](#), [2598](#), [2667](#), [2689](#),
 [2921](#), [3173](#), [3934](#), [4397](#), [4625](#), [4737](#), [4742](#),
 [5148](#), [5283](#), [5407](#), [5594](#), [5625](#), [5653](#), [5736](#), [5862](#)
 \MT@res@a [810](#), [812](#), [881](#), [888](#), [891](#), [893](#), [897](#), [902](#)
 \MT@res@b [811](#), [812](#), [891](#), [892](#), [893](#), [901](#), [902](#)
 \MT@reset@context [3994](#), [3998](#), [4004](#), [4031](#)
 \MT@reset@context@ [4004](#), [4031](#)
 \MT@reset@ef@codes [2392](#), [2416](#), [2444](#), [2497](#)
 \MT@reset@ef@codes@ [2439](#), [2449](#), [2453](#)
 \MT@reset@kn@codes [2603](#), [2610](#), [2640](#)
 \MT@reset@kn@codes@ [2640](#)
 \MT@reset@pr@codes [1664](#), [1682](#), [1693](#)
 \MT@reset@pr@codes@ [1693](#)
 \MT@reset@sp@codes [2525](#), [2532](#), [2570](#)
 \MT@reset@sp@codes@ [2570](#)
 \MT@restore@catcodes
 [5](#), [7](#), [8](#), [387](#), [393](#), [1295](#), [1302](#), [5949](#)
 \MT@restore@p@h [1104](#), [1135](#), [1200](#)
 \MT@saved@setupfont [5453](#), [5461](#)
 \MT@scale [1001](#), [1763](#), [1826](#), [1840](#), [1843](#), [2458](#)
 \MT@scale@factor [1765](#), [1796](#), [1824](#), [2020](#), [2515](#)
 \MT@scale@to@em [1728](#), [1738](#), [1760](#), [1793](#), [2026](#),
 [2538](#), [2545](#), [2552](#), [2616](#), [2623](#), [2849](#), [2972](#), [3143](#)
 \MT@scrubfeature [1516](#), [1528](#), [4546](#)
 \MT@scrubfeatures [1528](#), [4125](#), [4572](#)
 \MT@selectedfalse [543](#)
 \MT@selectedtrue [543](#)
 \MT@series [1512](#), [1628](#), [3376](#), [3388](#)
 \MT@set@all@ex [2439](#), [2516](#)
 \MT@set@all@kn [2640](#), [2662](#)
 \MT@set@all@pr [1685](#), [1693](#), [2016](#)
 \MT@set@all@sp [2570](#), [2595](#)
 \MT@set@babel@context
 [5877](#), [5903](#), [5908](#), [5922](#), [5927](#), [5929](#)
 \MT@set@codes [1670](#), [1704](#), [2398](#), [2531](#), [2609](#), [3281](#)
 \MT@set@curr@ls [2759](#)
 \MT@set@curr@ok [2779](#), [2781](#), [2787](#), [2790](#), [3027](#), [3031](#)
 \MT@set@curr@os [2761](#), [2789](#), [3027](#), [3031](#)
 \MT@set@default@set [4315](#), [4320](#), [4328](#)
 \MT@set@ex@codes [2420](#), [5668](#), [5670](#)
 \MT@set@ex@codes@n [2403](#), [2420](#), [5670](#)
 \MT@set@ex@codes@s [2388](#), [5668](#)
 \MT@set@ex@heirs [2471](#), [2508](#)
 \MT@set@inh@list [4768](#), [4770](#)
 \MT@set@inputenc [1666](#), [1919](#), [2394](#), [2527](#), [2605](#), [3449](#)
 \MT@set@inputenc@ [1922](#), [1924](#)
 \MT@set@kn@codes [2600](#)
 \MT@set@kn@heirs [2629](#), [2633](#)
 \MT@set@listname [1668](#),
 [1851](#), [2396](#), [2482](#), [2529](#), [2607](#), [2835](#), [3799](#)

- `\MT@set@lsbasefont` 2907, 2910
`\MT@set@lsfont` 2756, 2863, 2876
`\MT@set@named@keys` 4436, 4452, 4474, 4495, 4509, 4518
`\MT@set@pr@codes` 46, 1644
`\MT@set@pr@heirs` 1753, 1946
`\MT@set@pr@prefixes` 1958, 1963
`\MT@set@pr@prefixes@` 186, 200, 1963
`\MT@set@pr@prefixheirs` 344, 1677, 1953
`\MT@set@sp@codes` 2522
`\MT@set@sp@heirs` 2558, 2562
`\MT@set@tr@codes` 44, 76, 2684, 2871, 2881
`\MT@set@tr@zero` 2707, 2910, 5435
`\MT@SetExpansion` 4448, 4450
`\MT@SetExtraKerning` 4505, 4507
`\MT@SetExtraSpacing` 4491, 4493
`\MT@SetProtrusion` 4432, 4434
`\MT@SetTracking` 4470, 4472
`\MT@setup@` 248, 1051, 1055, 5140, 5141, 5944
`\MT@setup@contexts` 3993, 4011
`\MT@setup@copies` 5527, 5543
`\MT@setup@expansion` 5530, 5567
`\MT@setup@spacing@kerning` 5534, 5782, 5825
`\MT@setup@noligatures` 5535, 5860
`\MT@setup@PDF` 5526, 5538, 5548
`\MT@setup@protrusion` 5529, 5551
`\MT@setup@spacing` 5533, 5749, 5826
`\MT@setup@spacing@check` 5767, 5947, 5948
`\MT@setup@tracking` 5531, 5735, 5797, 5824
`\MT@setup@warntracking` 5532, 5829
`\MT@setupfont` 42, 90,
 165, 244, 1420, 3900, 5454, 5457, 5461, 5520
`\MT@setupfont@hook` 30, 252,
 1123, 1153, 1154, 1161, 1166, 1200, 1207, 1423
`\MT@shape` 1512, 1628, 3377, 3389
`\MT@shorthandoff` 5886, 5917, 5920
`\MT@show@pdfannot` 112
`\MT@shrink` 560, 2411,
 5590, 5591, 5598, 5599, 5602, 5606, 5673, 5686
`\MT@shrink@` 2411, 2426, 2430, 2436, 2481
`\MT@shrink@default` 576
`\MT@size` 24, 914, 918, 919, 1503, 1512, 1628
`\MT@size@name` 24, 907, 3404
`\MT@SOUL@doword` 1218, 1219
`\MT@sp@cc@name` 2528, 2530, 2582, 4493
`\MT@sp@context` 4007, 4048, 5771
`\MT@sp@doc@contexts` 4048
`\MT@sp@factor` 551
`\MT@sp@factor@` 1852
`\MT@sp@inh@name` 2556, 2557, 2558
`\MT@sp@max` 563
`\MT@sp@min` 563
`\MT@sp@setname` 4297
`\MT@sp@split@val` 2534
`\MT@sp@unit` 557
`\MT@sp@unit@` 1860, 2585
`\MT@spacing` 1449, 2519, 5763
`\MT@spacingfalse` 547
`\MT@spacingtrue` 547
`\MT@split@codes` 1706, 1710
`\MT@split@name` 1428, 1512
`\MT@step` 560, 2412, 5593,
 5595, 5613, 5617, 5620, 5687, 5690, 5691, 5695
`\MT@step@` 2412, 2426, 2430, 2436, 2481
`\MT@stretch` 560, 2410, 5587,
 5588, 5591, 5598, 5600, 5605, 5608, 5672, 5686
`\MT@stretch@` 2410, 2426, 2430, 2436, 2481
`\MT@stretch@default` 576, 5588
`\MT@strip@prefix` 3732, 3753, 3789
`\MT@temp` 300, 301, 302, 304, 317, 318,
 320, 1688, 1689, 1690, 1691, 2132, 2134,
 2138, 2152, 2155, 2158, 2161, 2165, 2169,
 2173, 2178, 2185, 2186, 2187, 2191, 2195,
 2199, 2210, 2214, 2307, 2309, 2311, 2315,
 2318, 2320, 2323, 2327, 2330, 2333, 2339,
 2342, 2345, 2347, 2357, 2364, 2367, 2370,
 2572, 2573, 2574, 2575, 2576, 2642, 2643,
 2644, 2645, 2944, 2948, 2951, 2959, 2962,
 3019, 3022, 3028, 3032, 3035, 3038, 3043,
 3051, 3053, 3059, 3133, 3136, 3153, 4243,
 4268, 4846, 4864, 4867, 5292, 5297, 5491,
 5496, 5501, 5502, 5504, 5507, 5702, 5706, 5710
`\MT@tempbox` 538, 2070, 2083, 2084,
 2085, 2086, 2087, 2091, 2092, 2100, 2376, 2380
`\MT@tempencoding` 4911, 4912, 4922, 4928
`\MT@tempfamily` 4923
`\MT@tempseries` 4924
`\MT@tempshape` 4925
`\MT@tempsize`
 107, 116, 247, 4879, 4926, 4935, 4937, 4941
`\MT@test@ast` 2970, 3141, 4107, 4116
`\MT@text@microtypecontext` 4001
`\MT@text@ls` 54, 3094, 3095, 3097
`\MT@text@microtypecontext` 4001
`\MT@the@pr@code` 1695, 1729, 1739, 2753
`\MT@the@pr@code@tr` 1695, 2753
`\MT@tikz@setup` 1115, 1136, 1223
`\MT@tlist@break` 23, 869, 904, 926, 2148, 3365, 5025
`\MT@toks` 55, 256, 537,
 1815, 1846, 2040, 2132, 2134, 2155, 2158,
 2161, 2164, 2168, 2172, 2181, 2190, 2195,
 2201, 2202, 2210, 2215, 2216, 2323, 2325,
 2339, 2342, 2345, 2347, 2360, 2367, 2375,
 2378, 2383, 2477, 3462, 3497, 3524, 3599,
 3606, 3657, 3745, 3760, 3769, 3804, 3816, 3824
`\MT@tr@cc@name` 2837,
 2838, 2845, 2846, 2859, 2860, 4475, 4482
`\MT@tr@context` 4007, 4048
`\MT@tr@doc@contexts` 4048
`\MT@tr@factor@` 2836
`\MT@tr@font@list` 2670
`\MT@tr@ispace` 2727, 2728, 2853
`\MT@tr@ligatures` 2743, 2856, 2923, 2926
`\MT@tr@max` 563, 3119, 3120, 3121
`\MT@tr@min` 563, 3123, 3124, 3125
`\MT@tr@noligatures` 2743, 2920
`\MT@tr@okern` 2741, 2855
`\MT@tr@ospace` 2730, 2731, 2853
`\MT@tr@outer@icr` 72, 3032, 3061
`\MT@tr@outer@icr@` 3061
`\MT@tr@outer@l` 2764, 2984
`\MT@tr@outer@next` 72, 3012, 3016, 3063
`\MT@tr@outer@r` 2791, 3012, 3028, 3035, 3061
`\MT@tr@outer@r@` 3013, 3018
`\MT@tr@set@okern` 2742, 3129
`\MT@tr@set@okern@` 3134, 3135, 3140
`\MT@tr@set@space` 2733, 2940
`\MT@tr@set@space@` 2945, 2946, 2947, 2953
`\MT@tr@set@space@@` 2956, 2961, 2964, 2969
`\MT@tr@setname` 4297
`\MT@tr@unit@` 2837, 2971, 3142

- `\MT@tracking` 125, 1436, 2670, 2871, 3902, 5435, 5436, 5744
`\MT@tracking@` 2670, 5436
`\MT@trackingfalse` 549, 5801
`\MT@trackingtrue` 549
`\MT@try@order` 3360, 3368
`\MT@undefined@char` 3646, 3651, 3701
`\MT@undo@patch` 1196, 1306, 5483
`\MT@upper` 4138, 4144, 4560
`\MT@use@set` 4284, 4289, 4297
`\MT@UseMicrotypeSet` 4280, 4282
`\MT@val` ... 4024, 4025, 4026, 4028, 4034, 4036, 4039, 4095, 4099, 4107, 4110, 4113, 4133, 4134, 4135, 4147, 4150, 4152, 4155, 4157, 4158, 4164, 4166, 4167, 4169, 4172, 4174, 4185, 4188, 4193, 4194, 4195, 4197, 4199, 4204, 4223, 4224, 4225, 4228, 4237, 4238, 4239, 4251, 4253, 4258, 4260, 4262, 4264, 4532, 4534, 4543, 4546, 4548, 4556, 4557, 4558, 4568, 4569, 4570, 4576, 4581, 4822, 4830, 4833, 4838, 4839, 5046, 5047, 5050, 5058, 5059, 5062, 5064, 5067, 5070, 5089, 5090, 5092, 5094, 5339, 5340, 5342, 5349, 5353, 5360, 5366, 5412, 5413, 5415, 5421, 5424
`\MT@variants` 3353, 4339, 4350
`\MT@version` 5266
`\MT@vinfo` 82, 99, 120, 1442, 1555, 1559, 1855, 1864, 1868, 1903, 2095, 2485, 2503, 2687, 2709, 3201, 3212, 3273, 3304, 3311, 3314, 3417, 3419, 3425, 4005, 4028, 4377, 4383, 5183, 5190, 5345, 5351, 5357, 5364, 5418, 5423, 5879
`\MT@warn@ascii` 3608, 3803
`\MT@warn@axis@empty` 4250, 4261, 4266
`\MT@warn@code@too@large` 46, 1830, 1833, 1837
`\MT@warn@err` 82, 5187, 5188
`\MT@warn@ex@too@large` 2461, 2464, 2475
`\MT@warn@maybe@inputenc` ... 3191, 3817, 3825, 3828
`\MT@warn@nodim` 4679, 4690, 4700, 4714
`\MT@warn@number@too@large` 3635, 3809
`\MT@warn@preset@towitz` ... 2009, 2029, 2586, 2655
`\MT@warn@rest` 3503, 3540, 3814
`\MT@warn@tracking@DVI` 2710, 5831
`\MT@warn@unknown` 3493, 3509, 3519, 3544, 3822
`\MT@warn@unknown@once` 3840, 3841
`\MT@warning` 82, 100, 1246, 1247, 1252, 1266, 1292, 1313, 3120, 3124, 3204, 3428, 4082, 4111, 4197, 4267, 4362, 4385, 4483, 4612, 4681, 4727, 4740, 4745, 4929, 4950, 5021, 5187, 5192, 5404, 5443, 5445, 5446
`\MT@warning@nl` 82, 94, 101, 244, 386, 515, 1043, 1047, 1083, 1095, 1172, 1807, 1844, 1896, 1936, 2030, 2476, 2692, 2932, 3804, 3810, 3815, 3823, 3872, 4455, 4654, 4855, 5032, 5039, 5050, 5132, 5188, 5193, 5227, 5263, 5271, 5307, 5321, 5472, 5517, 5574, 5618, 5674, 5693, 5756, 5772, 5837, 5931
- N** *Andy N* (contributor) U31, 253
Sven Naumann (contributor) U31, 249
newcomputermodern (package) 139
newpx (package) U21, 39, 139, 252
newtx (package) U21, 39, 139, 252
newunicodechar (package) 86, 252
Paolo Ney (contributor) U31, 253
nfssex (package) 56
- `\MT@while@num` 935, 961, 967, 1203, 1204, 1205
`\MT@with@babel@and@T` 1060, 1125, 1126, 1127, 1157, 1158, 1159, 1296, 1297, 5912, 5913, 5914, 5915, 5916, 5919
`\MT@with@package@T` 1057, 1108, 1110, 1129, 1132, 1133, 1134, 1136, 1141, 1148, 1149, 1150, 1151, 1152, 1164, 1180, 1194, 1199, 1201, 1223, 1224, 1238, 1368, 1413, 2254, 2256, 2267, 2291, 2295, 3667, 3683, 3684, 3840, 3841, 3883, 5755
`\MT@xadd` 110, 835, 3301, 3312, 3959, 3970, 4036, 4136, 4379, 4559, 4830, 4832
`\MT@xaddb` 843, 4579, 4939
`\MT@xdef@n` .. 643, 4299, 4302, 4331, 4336, 4365, 4482, 4574, 4596, 4756, 4794, 4960, 5047, 5049
`\MT@xetex@no` 482
`\MT@xspace` 3051, 3066
`\MT@xspace@` 3066
`\MT@xunicode@false` 1109
`\MT@xunicodetrue` 1109, 1110, 1151
`\MTS@Char` 165, 194
`\MTS@Char@do` 194, 198
`\MTS@charwd` 273
`\MTS@crulefill` 155, 217, 221
`\MTS@first` 238, 251, 265
`\MTS@firstorsecond` 241, 254, 264
`\MTS@font` 259, 261
`\MTS@glyphlist` 8, 168, 277, 294, 302, 318, 333
`\MTS@lede` 232, 237, 250, 257
`\MTS@load@list` 185, 3262
`\MTS@lp@` 273, 297, 1727, 1730, 1731, 1968, 1985
`\MTS@lpcode` 213, 1727, 1731, 1746
`\MTS@pr@split@val` 184, 1723
`\MTS@printtext` 155, 218, 220, 260, 298, 304, 315, 326, 328, 334, 341, 347, 1649, 1662, 1672, 1675, 1681, 1745, 1971, 3272, 3283
`\MTS@Prot` 165, 178
`\MTS@Prot@do` 178, 182
`\MTS@rp@` 273, 297, 1737, 1740, 1741, 1968, 1994
`\MTS@rpcode` 213, 1737, 1741, 1747
`\MTS@second` 239, 252, 267, 270
`\MTS@set@pr@prefixes@` 186, 200, 1966
`\MTS@setup` .. 168, 177, 193, 206, 210, 231, 236, 249
`\MTS@show@char` 292, 320, 334
`\MTS@show@char@pr` 273, 304, 343, 1744, 1754, 1971, 1998
`\MTS@show@char@x` 292, 307, 337, 343
`\MTS@show@index` 155, 219, 290, 293
`\MTS@show@inheritance` 201, 324
`\MTS@show@missing` 295, 1679
`\MTS@show@missing@` 312, 314, 345
`\MTS@show@pr` 187, 1645
`\MTS@show@protrusion@line` 207, 211, 213, 233, 243, 255
`\MTS@temp` 242, 255, 265, 267, 270
Peter Muthesius (contributor) U31, 245
- nfssex (package) 56, 255
NightShade (contributor) U31, 255
nopatch (option) U6, U26, U29, 121, 254
\noprotrusion U25, 52, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343, 2035
\noprotrusionifhmode U25, 53, 2038
\normalfont 1419, 2232, 5713, 5717
\nullfont 1419

- O** *Heiko Oberdiek* (contributor) [U31](#), [251](#)
Phelype Oleinik (contributor) [U31](#), [253](#)
- P** *Scott Pakin* (contributor) [U31](#), [251](#)
Giuseppe Palma (contributor) [U31](#)
patch (option) [U6](#), [U26](#), [16](#), [121](#), [254](#)
`\pdfadjustinterwordglue` [5437](#), [5753](#)
`\pdfadjustspacing` [434](#), [1219](#), [5396](#), [5400](#), [5715](#)
`\pdfappendkern` [5439](#), [5786](#)
`\pdfcopyfont` [1473](#)
`\pdfcpot` (package) [U28](#), [U32](#), [30](#), [45](#), [90](#), [244](#)
`\pdffontexpand` [2423](#), [2426](#), [2430](#), [2436](#)
`\pdffontsize` [1501](#), [1504](#)
`\pdfnoligatures` [3187](#), [3203](#), [3209](#)
`\pdfoutput` [400](#), [432](#), [5122](#), [5126](#),
[5127](#), [5129](#), [5130](#), [5282](#), [5540](#), [5569](#), [5626](#), [5834](#)
`\pdfprependkern` [5438](#), [5785](#)
`\pdfprotrudechars` [433](#), [5395](#), [5399](#), [5848](#)
`pdfTeX` (engine) [U1](#),
[U3](#), [U5](#), [U6](#), [U7](#), [U8](#), [U12](#), [U13](#), [U14](#), [U15](#),
[U16](#), [U17](#), [U18](#), [U19](#), [U23](#), [U24](#), [U26](#), [U27](#),
[U28](#), [U29](#), [U30](#), [U31](#), [U33](#), [U34](#), [U35](#), [1](#), [4](#), [5](#),
- Q** *qfonts* (package) [U21](#), [139](#), [247](#)
- R** *Lars Rönnbäck* (contributor) [U31](#), [251](#)
ragged2e (package) [U27](#), [71](#), [132](#), [250](#), [251](#)
rallg (contributor) [U31](#), [255](#)
RazorXsr (contributor) [U31](#), [251](#)
reledmac (package) [U30](#), [28](#), [252](#)
reledpar (package) [U30](#)
relesize (package) [100](#)
`\remove@tlig` [88](#), [3768](#)
`\rightmarginkern` [1079](#)
- S** *Safron* (contributor) [U31](#), [254](#)
Wolfram Schaalo (contributor) [U31](#), [250](#)
Ekkehart Schlicht (contributor) [U31](#), [253](#)
scholnik (contributor) [U31](#), [256](#)
Christopher Schramm (contributor) [U31](#), [252](#)
Sebastian Schubert (contributor) [U31](#), [251](#)
Torsten Schuetze (contributor) [U31](#), [254](#)
Herb Schulz (contributor) [U31](#)
Ulrich Schwarz (contributor) [U31](#), [20](#)
selected (option) [U7](#), [U13](#), [U35](#), [59](#), [80](#), [118](#), [245](#)
`\selectfont` [180](#), [189](#), [196](#), [203](#),
[227](#), [258](#), [2232](#), [2872](#), [3997](#), [4009](#), [5332](#), [5939](#)
`\seriesdefault` [226](#)
`\set@fontsize` [4204](#)
`\SetExpansion`
[U13](#), [36](#), [38](#), [47](#), [59](#), [104](#), [105](#), [110](#), [244](#), [4445](#)
`\SetExtraKerning` [U17](#), [U33](#), [38](#), [4503](#)
`\SetExtraSpacing` [U18](#), [U33](#), [39](#), [4488](#)
`\SetProtrusion` [U7](#), [U12](#),
[U14](#), [35](#), [38](#), [44](#), [104](#), [244](#), [246](#), [247](#), [248](#), [4430](#)
`\SetTracking` [U15](#), [U23](#), [U24](#), [U33](#), [37](#), [47](#), [65](#), [111](#), [4468](#)
`\sfcode` [5770](#)
`\shapedefault` [226](#)
shapepar (package) [U31](#), [29](#)
`\shbscode` [2553](#), [2554](#), [2565](#), [2568](#), [2575](#)
shortvrb (package) [86](#), [255](#)
`\Showbaselinecolor` [149](#), [218](#), [220](#), [280](#)
`\ShowCharacterInheritance` [U26](#), [191](#)
`\ShowDummyLine` [223](#), [225](#), [262](#)
`\ShowGlyphIndexfalse` [149](#)
`\ShowGlyphIndextrue` [U26](#), [149](#)
`\showhyphens` [5701](#)
`\ShowMissingGlyphsfalse` [150](#)
- `\outputmode` [432](#), [1502](#), [5122](#)
[7](#), [13](#), [14](#), [15](#), [17](#), [18](#), [21](#), [28](#), [37](#), [38](#), [39](#), [45](#),
[60](#), [62](#), [63](#), [64](#), [65](#), [69](#), [70](#), [73](#), [75](#), [104](#), [106](#),
[108](#), [109](#), [110](#), [122](#), [126](#), [128](#), [129](#), [130](#), [133](#),
[198](#), [200](#), [244](#), [245](#), [246](#), [247](#), [248](#), [249](#), [250](#), [254](#)
pdftexcmds (package) [7](#), [250](#)
`\pdftracingfonts` [595](#)
Wiebke Petersen (contributor) [U31](#), [249](#)
`\pickup@font` [3871](#), [3873](#), [3882](#), [3906](#), [3907](#), [3908](#)
pifont (package) [91](#)
pinyin (package) [32](#), [249](#), [250](#)
Paolo Polesana (contributor) [U31](#), [253](#)
polyglossia (package) [U8](#), [U23](#), [U28](#), [135](#), [255](#)
`\protrudechars` [433](#), [5395](#)
protrusion (option) [U5](#), [U10](#), [117](#), [118](#)
psnfss (package) [69](#), [199](#), [252](#)
pstricks (package) [128](#)
David Purton (contributor) [U31](#), [255](#)
pxfonts (package) [U21](#), [139](#)
- `\rightprotrusion`
. [U25](#), [U32](#), [51](#), [1358](#), [1360](#), [1364](#), [1377](#), [2058](#)
Will Robertson (contributor) [U31](#), [252](#)
Nathan Rosenblum (contributor) [U31](#), [250](#)
Mark Rossi (contributor) [U31](#), [246](#)
Maïeul Rouquette (contributor) [U31](#), [251](#)
Colinourke (contributor) [U31](#), [250](#)
Élie Roux (contributor) [U30](#), [251](#)
`\rptide` [215](#), [269](#), [1690](#),
[1739](#), [1742](#), [1948](#), [1951](#), [1993](#), [1994](#), [1997](#), [2752](#)
- `\ShowMissingGlyphstrue` [U26](#), [150](#)
`\Shownegcolor` [149](#), [282](#), [286](#)
`\Showposcolor` [149](#), [282](#), [286](#)
`\ShowProtrusion` [U26](#), [175](#)
`\ShowProtrusionAll` [U26](#), [230](#)
`\ShowProtrusionDefined` [U26](#), [235](#)
`\ShowProtrusionLineGlyph` [205](#)
`\ShowProtrusionLineIndex` [209](#)
`\ShowProtrusionMissing` [U26](#), [248](#)
shrink (option) [U7](#), [U14](#), [U35](#), [120](#), [129](#), [131](#), [247](#), [249](#)
Uwe Siart (contributor) [U31](#), [256](#)
simon-codes-something (contributor) [U31](#)
simplecv (class) [U25](#), [34](#)
slantsc (package) [177](#)
soul (package) [U3](#), [U32](#), [31](#), [72](#), [248](#), [250](#), [253](#)
`\spacefactor` [2995](#)
`\spaceskip` [2996](#), [2999](#)
spacing (option) [U6](#), [U10](#), [U27](#), [U33](#), [118](#), [250](#), [251](#)
Christian Stark (contributor) [U31](#), [251](#)
`\stbscode` [2546](#), [2547](#), [2564](#), [2568](#), [2574](#)
Ralf Steinle (contributor) [U31](#), [255](#)
step (option) [U7](#), [U33](#), [U35](#), [120](#), [129](#), [131](#), [245](#), [247](#), [249](#)
step (package) [U21](#), [139](#), [253](#)
stickstoo (package) [U21](#), [139](#), [255](#)
Stephan Stiller (contributor) [U31](#)
stix (package) [U21](#), [139](#), [253](#)
stix2 (package) [U21](#), [139](#), [253](#)
Linas Stonys (contributor) [U31](#), [256](#)
stretch (option) [U7](#), [U14](#), [U27](#), [U35](#), [120](#), [129](#), [131](#), [249](#)
Ralf Stubner (contributor) [U31](#)
svjour3 (class) [100](#), [253](#)
syntax (package) [254](#)

- T** `\tagcode` 2747, 3190, 3196
`tempora` (package) U21, 139, 252
`tex4ht` (package) 30, 248, 255
`TEX Live` (distribution) U30, 13, 140, 247
`\textls` U15,
U23, U33, 31, 57, 58, 65, 67, 69, 70, 72, 74,
248, 251, 1188, 1216, 2767, 2867, 2884, 3092
`\textls*` U23
`\textmicrotypecontext` ... U22, U33, 49, 1184, 4001
Hàn Thế Thành (contributor) U30, 245, 247, 251
theufman (contributor) U31, 255
`tikz` (package) 29, 240, 251, 252
- U** `ucs` (package) 86, 250
Holger Uhr (contributor) U31, 247
`ulgothic` (package) U21
uli (contributor) U31, 252
`unicode-math` (package) 32, 254
`unit` (option) U7, U34, 246, 247, 248
`\UseMicrotypeSet` U10, 33, 65, 79, 4076, 4278
user11126 (contributor) U31, 252
- V** *Geoff Vallis* (contributor) U31, 249
`verbatim` (package) 36
`verbose` (option) U8, U33, U34, 120, 246, 249
Georg Verweyen (contributor) U31, 245, 247
- W** *Liang-Bo Wang* (contributor) U31, 254
Moritz Wemheuer (contributor) U31, 253
Franz Wexler (contributor) U31, 253
Peter Wilson (contributor) U30, 28
- X** `XCharter` (package) U21, 140, 252
`xeCJK` (package) U28, 91, 252
`XYTEX` (engine) U1, U3, U5, U6,
U8, U11, U12, U19, U28, U30, U32, U33, 1,
4, 13, 15, 26, 28, 29, 37, 39, 49, 73, 83, 126,
131, 133, 138, 141, 145, 154, 223, 251, 252, 253
`\XeTeXcharglyph` 3530
`\XeTeXfonttype` 3481
`\XeTeXglyph` 1777
- Y** `yfonts` (package) U24, 57
- Z** *Michael Zedler* (contributor) U31
- `tikzposter` (class) 252
`tipa` (package) 91
`titletoc` (package) 35
tnull (contributor) U31, 254
`tocbasic` (package) 34
`trace` (package) 92, 248
`\tracingmicrotype` 6, 93
`\tracingmicrotypeinpdf` 7, 104
`\tracingmicrotypeinpdfallfalse` 7, 111
`\tracingmicrotypeinpdfalltrue` 111
`tracking` (option) U5, U10, U23, U33, 118
Antonis Tsolomitis (contributor) U30, 202, 254
`txfonts` (package) U21, 139
user182849 (contributor) U31, 255
`\UTF@four@octets@noexpand` 3649
`\UTF@three@octets@noexpand` 3648
`\UTF@two@octets@noexpand` 3647
`\UTFviii@defined` 3650
`\UTFviii@four@octets` 2344
`\UTFviii@three@octets` 2341
`\UTFviii@two@octets` 2338
Bruno Victal (contributor) U31, 256
Herbert Voß (contributor) U31, 247, 252
Hendrik Vogt (contributor) U30, 160, 251
Maverick Woo (contributor) U31, 250
`wordcount` (package) 27, 249
Joseph Wright (contributor) U31
- `\XeTeXglyphindex` 1974, 1975, 3517
`\XeTeXglyphname` 3536
`\XeTeXlastfontchar` 990
`\XeTeXprotrudechars` 5403
`\xspace` 3050, 3051
`xspace` (package) 73, 251
`\xspaceskip` 2998, 3001
`xunicode` (package) 88, 89, 251
`xunicode-addon` (package) 91
Akira Yokosawa (contributor) U31, 255
`zefonts` (package) U21, 138

E The L^AT_EX Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 L^AT_EX3 Project

Preamble

The L^AT_EX Project Public License (LPPL) is the primary license under which the L^AT_EX kernel and the base L^AT_EX 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 T_EX-related (such as a L^AT_EX package), but it is written in such a way that you can use it even if your work is unrelated to T_EX.

The section ‘WHETHER AND HOW TO DISTRIBUTE

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

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 L^AT_EX3 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 L^AT_EX 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 ‘L^AT_EX-Format’ or in the case of files belonging to the ‘L^AT_EX-format’ a program implementing the ‘T_EX 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 au-

thor 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 L^AT_EX 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 be-

comes 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 L^AT_EX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L^AT_EX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L^AT_EX, 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 ‘L^AT_EX-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.