

# The microtype package

Subliminal refinements towards typographical perfection

R Schlicht  
w.m.1@gmx.net

v2.8b  
2021/02/25

---

The microtype package provides a  $\LaTeX$  interface to the micro-typographic extensions that were introduced by pdf $\TeX$  and have since also propagated to Lua $\TeX$  and X $\TeX$ : 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 pdf $\TeX$  (version 0.14f or later), Lua $\TeX$ , or X $\TeX$  (at least version 0.9997). Font expansion works with pdf $\TeX$  (version 1.20 for automatic expansion) or Lua $\TeX$ . The package will by default enable protrusion and expansion if they can safely be assumed to work. Disabling ligatures requires pdf $\TeX$  ( $\geq 1.30$ ) or Lua $\TeX$ , while the adjustment of interword spacing and of kerning only works with pdf $\TeX$  ( $\geq 1.40$ ). Letterspacing is available with pdf $\TeX$  ( $\geq 1.40$ ) or Lua $\TeX$  ( $\geq 0.62$ ).

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).

This package is copyright © 2004–2021 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 ‘author-maintained’.

## Contents

<b>1</b>	<b>Micro-typography with T<sub>E</sub>X</b>	<b>4</b>
<b>2</b>	<b>Getting started</b>	<b>5</b>
<b>3</b>	<b>Options</b>	<b>6</b>
3.1	Enabling the micro-typographic features . . . . .	6
3.2	Character protrusion . . . . .	7
3.3	Font expansion . . . . .	7
3.4	Tracking . . . . .	8
3.5	Miscellaneous options . . . . .	9
3.6	Changing options later . . . . .	9
<b>4</b>	<b>Selecting fonts for micro-typography</b>	<b>10</b>
<b>5</b>	<b>Micro fine tuning</b>	<b>12</b>
5.1	Character protrusion . . . . .	13
5.2	Font expansion . . . . .	14
5.3	Tracking . . . . .	15
5.4	Additional kerning . . . . .	18
5.5	Interword spacing . . . . .	19
5.6	Character inheritance . . . . .	20
5.7	Configuration files . . . . .	20
<b>6</b>	<b>Context-sensitive setup</b>	<b>22</b>
<b>7</b>	<b>Letterspacing revisited</b>	<b>23</b>
<b>8</b>	<b>Disabling ligatures</b>	<b>24</b>
<b>9</b>	<b>Hints and caveats</b>	<b>25</b>
<b>10</b>	<b>Contributions</b>	<b>28</b>
<b>11</b>	<b>Acknowledgments</b>	<b>28</b>
<b>12</b>	<b>References</b>	<b>30</b>
<b>13</b>	<b>Short history</b>	<b>30</b>
<b>14</b>	<b>Implementation</b>	<b>34</b>
14.1	Preliminaries . . . . .	35
	Debugging [36] Requirements [38] Declarations [42] Auxiliary macros [43] Compatibility [52]	
14.2	Font setup . . . . .	56
	Protrusion [62] Expansion [69] Interword spacing (glue) [72] Additional kern- ing [73] Tracking [74] Disabling ligatures [85] Loading the configuration [87] Translating characters into slots [91] Hook into L <sup>A</sup> T <sub>E</sub> X's font selection [99] Context-sensitive setup [102]	
14.3	Configuration . . . . .	104
	Font sets [105] Variants and aliases [111] Disabling ligatures [112] Interaction with babel [112] Fine tuning [112] Character inheritance [119] Permuta- tion [121]	

14.4	Package options . . . . .	124
	Declaring the options [124] Loading the definition file [128] Reading the configuration file [129] Hook for other packages [130] Changing options later [130] Processing the options [133]	
<b>15</b>	<b>Configuration files</b>	<b>143</b>
15.1	Font sets . . . . .	143
15.2	Font variants and aliases . . . . .	144
15.3	Interaction with babel . . . . .	146
15.4	Note on admissible characters . . . . .	147
15.5	Character inheritance . . . . .	147
	OT1 [147] T1 [148] LY1 [148] OT4 [149] QX [149] T5 [150] EU1, EU2, TU [151] Euro symbols [151]	
15.6	Tracking . . . . .	152
15.7	Font expansion . . . . .	152
15.8	Character protrusion . . . . .	154
	Normal [155] Italics [163] Small caps [174] Italic small caps [177] Text companion [178] Computer Modern math [183] AMS symbols [187] Euler [190] Euro symbols [194]	
15.9	Interword spacing . . . . .	195
	Nonfrenchspacing [197]	
15.10	Additional kerning . . . . .	198
	French [199] Turkish [199]	
<b>16</b>	<b>OpenType configuration files</b>	<b>201</b>
16.1	Character inheritance . . . . .	201
16.2	Character protrusion . . . . .	205
<b>17</b>	<b>Auxiliary file for micro fine tuning</b>	<b>219</b>
<b>A</b>	<b>The title logo</b>	<b>221</b>
A.1	Macros . . . . .	221
A.2	Document . . . . .	226
<b>B</b>	<b>The letterspacing illustration</b>	<b>226</b>
B.1	Macros . . . . .	226
B.2	Document . . . . .	229
<b>C</b>	<b>Change history</b>	<b>231</b>
<b>D</b>	<b>Index</b>	<b>240</b>
<b>E</b>	<b>The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>249</b>

## List of Tables

1	Availability of micro-typographic features . . . . .	7
2	Predefined font sets . . . . .	11
3	Fonts with tailored protrusion settings . . . . .	21
4	Order for matching font attributes . . . . .	89

## 1 Micro-typography with T<sub>E</sub>X

Micro-typography is the art of enhancing the appearance and readability of a document while exhibiting a minimum degree of visual obtrusion. It is concerned with what happens between or at the margins of characters, words or lines. Whereas the macro-typographical aspects of a document (i.e., its layout) are clearly visible even to the untrained eye, micro-typographical refinements should ideally not even be recognisable. That is, you may think that a document looks beautiful, but you might not be able to tell exactly why: good micro-typographic practice tries to reduce all potential irritations that might disturb a reader.

Some essential micro-typographical aspects are already taken care of by T<sub>E</sub>X out of the box – and in an outstanding manner – namely, hyphenation and justification, as well as kerning and ligatures. Other aspects are in the user’s scope of responsibilities, e.g., to specify the right amounts of spacing around punctuation characters, numbers, or quotation marks. On top of this, a number of long-standing micro-typographic techniques have been introduced to the T<sub>E</sub>X world relatively recently with pdfT<sub>E</sub>X, and have since also propagated to LuaT<sub>E</sub>X and XeT<sub>E</sub>X. These features make them the tool of choice not only for the creation of electronic documents but also of works of outstanding time-honoured typography: most prominently, *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Thế Thành, the author of pdfT<sub>E</sub>X, who writes in his thesis:

After you have read the text on the right, you can view the effect of the features it describes by clicking on the links:

Protrusion	off
Expansion	off

Both features are enabled throughout this document.

‘Margin kerning is the adjustments of the characters at the margins of a typeset text. A simplified employment of margin kerning is hanging punctuation. Margin kerning is needed for optical alignment of the margins of a typeset text, because mechanical justification of the margins makes them look rather ragged. Some characters can make a line appear shorter to the human eye than others. Shifting such characters by an appropriate amount into the margins would greatly improve the appearance of a typeset text.

Composing with font expansion is the method to use a wider or narrower variant of a font to make interword spacing more even. A font in a loose line can be substituted by a wider variant so the interword spaces are stretched by a smaller amount. Similarly, a font in a tight line can be replaced by a narrower variant to reduce the amount that the interword spaces are shrunk by. There is certainly a potential danger of font distortion when using such manipulations, thus they must be used with extreme care. The potentiality to adjust a line width by font expansion can be taken into consideration while a paragraph is being broken into lines, in order to choose better breakpoints.’ [Thành 2000, p. 323]

Another micro-typographic technique, which has always been extremely difficult to achieve in T<sub>E</sub>X, is robust and hyphenatable *letterspacing* (*tracking*).<sup>1</sup> Whereas letterspacing can easily be, and often is, abused when applying it to lowercase letters, readability may be increased by slightly letterspacing (small) capitals or by decreasing the tracking of very large uppercase type.

Setting *additional kerning* for individual characters is especially (but not only) useful for languages whose typographical tradition requires certain characters to be separated by a space. For example, it is customary in French typography to add a small space before question mark, exclamation mark and semi-colon, and a bigger space before the colon and the guillemets. Until now, this could only be achieved

<sup>1</sup> The `soul` package undertakes great efforts, but may still fail in certain circumstances; even to systematically adjust the tracking of a font throughout the document remains impossible.

by making these characters active (as is done, for example, by the `babel` package), which may not always be a robust solution. In contrast to the standard kerning built into the fonts (which will of course apply as usual), this additional kerning relates to single characters, not to character pairs.

*Adjustment of interword spacing* is based upon the idea that in order to achieve a uniform greyness of the text, the space between words should also depend on the surrounding characters. For example, if a word ends with an ‘r’, the following space should be a tiny bit smaller than that following, say, an ‘m’. You can think of this concept as an extension to TeX’s ‘space factors’. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently influence the interword space. Also, the settings shipped with `microtype` are but a first approximation, and I would highly welcome corrections and improvements. I suggest reading the reasoning behind the settings in section 15.9.

The possibility, finally, to *disable all or selected ligatures* is particularly useful for typewriter fonts.

The `microtype` package provides an interface to all these micro-typographic extensions. All micro-typographic aspects may be customised to your taste and needs in a straight-forward and systematic manner. The next chapters present a survey of all options and customisation possibilities. Should the micro-typographic extension discussed in a section work only with certain TeX engines, this requirement is marked inside a grey text box on the right.

## 2 Getting started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

This will be sufficient in most cases, and if you are not interested in fine-tuning the micro-typographic appearance of your document (however unlikely this would seem, since using this package is proof of your interest in typographic issues), you may actually skip the rest of this document. If this, on the other hand, does not satisfy you – be it for theoretical or practical reasons – this manual will guide you on the path to the desired results along the following milestones:

- Enable the desired micro-typographic features, either via the respective package option or with the `\microtypesetup` command (section 3).
- Select the fonts to which this feature should be applied by declaring and activating ‘sets of fonts’. A number of sets are predefined, which may be activated directly in the package options (section 4).
- Fine-tune the micro-typographic settings of the fonts or sets of fonts (section 5).
- If you’re of the kind who always wants to march on, you will certainly be interested in the possibility of context-sensitive setup (section 6).
- You are even countenanced to leave the path of typographic virtue and steal some sheep (section 7) or trespass in other ways (section 8).
- Should you encounter any obstacles, follow the hints and caveats (section 9).

## 3 Options

Like many other  $\text{\LaTeX}$  packages, the `microtype` package accepts options in the well-known key=value syntax. In the following, you will find a description of all **keys** and their possible values (‘true’ may be omitted; multiple values, where allowed, must be enclosed in braces; the default value is shown on the right – if preceded by an asterisk, this default only applies when running an up-to-date  $\text{\pdfTeX}$  in PDF mode).

### 3.1 Enabling the micro-typographic features

**protrusion** true, false, compatibility, nocompatibility, *<font set name>* \*true

**expansion** These are the main options to control the level of micro-typographic refinement which the fonts in your document should gain. By default, the package is moderately greedy: character protrusion will always be enabled, font expansion will only be disabled when the fonts cannot be expanded automatically, that is, with  $\text{\pdfTeX}$  versions older than 1.20 or in DVI output mode (see section 3.5), or with  $\text{\XeTeX}$ . In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (hence, it is usually not necessary to load the package with different options, e.g., for PDF resp. DVI mode).

**activate** Protrusion and expansion may be enabled or disabled independently from each other by setting the respective key to true resp. false. The `activate` option is a shortcut for setting both options at the same time. Therefore, the following lines all have the same effect (when creating PDF files with a recent version of  $\text{\pdfTeX}$ ):

```
\usepackage[protrusion=true,expansion]{microtype}
```

```
\usepackage[activate={true,nocompatibility}]{microtype}
```

```
\usepackage{microtype}
```

With activated font expansion and/or character protrusion, line breaks (and consequently, page breaks) may turn out differently. If this is not desired – because you are re-typesetting a book whose pagination must not change – you may pass the value `compatibility` to the `protrusion` and/or `expansion` options. Typographically, however, the results will be suboptimal, hence the default value is `nocompatibility`.

Finally, you may also specify the name of a font set to which character protrusion and/or font expansion should be restricted. See section 4 for a detailed discussion. Specifying a font set for a feature implicitly activates this feature.

**tracking** true, false, *<font set name>* false

This option will systematically change the tracking of the fonts specified in the active font set (by default, all small capitals). It is not available with  $\text{\XeTeX}$  (you may use the ‘LetterSpace’ option of the `fontspec` package instead). With  $\text{\pdfTeX}$ , it is only available in PDF mode.

**kerning** true, false, *<font set name>* false

**spacing** These features do not unconditionally improve the quality of the typeset text: the spacing feature is still considered experimental, while the kerning feature only makes sense in special cases. Therefore, neither feature is enabled by default. They are not available with  $\text{\XeTeX}$  or  $\text{\LuaTeX}$ .

Table 1:

Availability of micro-typographic features

T <sub>E</sub> X engine			Micro-typographic features					
Engine	Version	Output	Protrusion	Expansion		Tracking	Kerning	Spacing
				manual	automatic			
pdfT <sub>E</sub> X	< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
	≥ 0.14f	DVI/PDF	★	☒	∅	∅	∅	∅
	≥ 1.20	DVI	★	☒	∅	∅	∅	∅
		PDF	★	☒	★	∅	∅	∅
	≥ 1.40	DVI	★	☒	∅	∅	☒	☒
		PDF	★	☒	★	☒	☒	☒
LuaT <sub>E</sub> X	≥ 0.30	DVI	★	☒	∅	∅	∅	∅
		PDF	★	☒	★	∅	∅	∅
	≥ 0.62	DVI	★	∅	(☒) <sup>a</sup>	☒	∅	∅
		PDF	★	∅	★	☒	∅	∅
X <sub>Y</sub> T <sub>E</sub> X	≥ 0.9997	PDF	★	∅	∅	∅	∅	∅
★ = enabled    ☒ = not enabled    ∅ = not available <sup>a</sup> by means of variable tracking								

Table 1 presents an overview of which micro-typographic features are available and enabled by default for the relevant T<sub>E</sub>X versions and output modes.

Whether ligatures should be disabled cannot be controlled via a package option but by using the `\DisableLigatures` command, which is explained in section 8.

3.2 Character protrusion

pdfT<sub>E</sub>X 0.14f | LuaT<sub>E</sub>X 0.30 | X<sub>Y</sub>T<sub>E</sub>X 0.9997

**factor** *<integer>* 1000

Using this option, you can globally increase or decrease the amount by which the characters will be protruded. While a value of 1000 means that the full protrusion as specified in the configuration (see section 5.1) will be used, a value of 500 would result in halving all protrusion factors of the configuration. This might be useful if you are generally satisfied with the settings but prefer the margin kerning to be less or more visible (e.g., if you are so proud of being able to use this feature that you want everybody to see it, or – to mention a motivation more in compliance with typographical correctness – if you are using a large font that calls for more modest protrusion).

**unit** character, *<dimension>* character

This option is described in section 5.1, apropos the command `\SetProtrusion`. Use with care.

3.3 Font expansion

pdfT<sub>E</sub>X 0.14f | LuaT<sub>E</sub>X 0.30

**auto** true, false \*true

Beginning with pdfT<sub>E</sub>X version 1.20 (inherited by LuaT<sub>E</sub>X), the expanded instances of the fonts may be calculated automatically and at run-time instead of the user

having to prepare them in advance. This option is true by default provided that you are using a TeX engine with this capability and the output mode is PDF.<sup>2</sup> If auto is set to false, the font instances for all expansion steps must exist (with files called *(font name)±(expansion value)*, e.g., *cmr12+10*, as described in the [pdfTeX manual](#)). With recent versions of LuaTeX, expansion is always automatic.

When generating DVI files, font expansion has to be enabled explicitly. With pdfTeX, *automatic* font expansion will not work because the postprocessing drivers (dvips, dvipdfm, etc.) resp. the DVI viewer are not able to generate the fonts on the fly. With LuaTeX, on the other hand, expansion in DVI mode is realised by modifying the inter-letter spacing (tracking) instead of the glyphs themselves, which may or may not be desired.

**stretch** *(integer)* 20

**shrink** You may specify the stretchability and shrinkability of a font, i.e., the maximum amount that a font may be stretched or shrunk. The numbers will be divided by 1000, so that a stretch limit of 10 means that the font may be expanded by up to 1%. The default stretch limit is 20. The shrink limit will by default be the same as the stretch limit.

**step** *(integer)* \* 1

Fonts are not expanded by arbitrary amounts but only by certain discrete steps within the expansion limits. With recent versions of pdfTeX (1.40 or newer) or LuaTeX, this option is by default set to 1, in order to allow trying the maximum number of font instances, and hence to guarantee the best possible output.<sup>3</sup> Older pdfTeX versions, however, had to include every font instance in the PDF file, which may increase the file size quite dramatically. Therefore, in case you are using a pre-1.40 pdfTeX version, step is by default set to one fifth of the smaller value of stretch and shrink.

**selected** true, false false

When applying font expansion, it is possible to restrict the expansion of some characters that are more sensitive to deformation than others (e.g., the ‘O’, in contrast to the ‘I’). This is called *selected expansion*, and its usage allows increasing the stretch and shrink limits (to, say, 30 instead of 20); however, the gain is limited since at the same time the average stretch variance will be decreased. Therefore, this option is by default set to false, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.

### 3.4 Tracking

pdfTeX 1.40 | LuaTeX 0.62

**letterspace** *(integer)* 100

This option changes the default amount for tracking (see section 5.3) resp. letter-spacing (see section 7). The amount is specified in thousandths of 1em; admissible values are in the range of −1000 to +1000.

- 2 With pdfTeX, automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding, you should either install the *cm-super* package or use the Latin Modern fonts (package *lmodern*).
- 3 The downside with this default is that pdfTeX may run out of memory with huge documents; in this case, read about the error messages in the ‘Hints and caveats’ section (9), or try with a larger step.



### 3.5 Miscellaneous options

**draft** true, false false

**final** If the `draft` option is passed to the package, *all micro-typographic extensions will be disabled*, which may lead to different line, and hence page, breaks. The `draft` and `final` options may also be inherited from the class options; of course, you can override them in the package options. E.g., if you are using the class option `draft` to show any overfull boxes, you should load `microtype` with the `final` option.

**verbose** true, false, errors, silent false

Information on the settings used for each font will be written into the log file if you enable the `verbose` option. When `microtype` encounters a problem that is not fatal (e.g., an unknown character in the settings, or non-existent settings), it will by default only issue a warning and try to continue. Loading the package with `verbose=errors` will turn all warnings into errors, so that you can be sure that no problem will go unnoticed. If on the other hand you have investigated all warnings and decide to ignore them, you may silence `microtype` with `verbose=silent`.

**babel** true, false false

Loading the package with the `babel` option will adjust the typesetting according to the respective selected language. Read section 6 for further information.

**config** *<file name>* `microtype`

Various settings for this package will be loaded from a main configuration file, by default `microtype.cfg` (see section 5.7). You can have a different configuration file loaded instead by specifying its name *without the extension*, e.g., `config=microtype`.

**DVIoutput** true, false \* false

`pdfTeX` and `LuaTeX` are not only able to generate PDF output but can also spit out DVI files. In fact, all recent `TeX` systems are using `pdfTeX` as the default engine also for DVI output, and `LuaTeX` too can be called in DVI mode. However, since changing the output mode inside the document may have undesired effects, this option should be considered deprecated; instead, it is recommended to just call the respective program (`latex` resp. `dvilualatex`). For `XYTeX`, this option is not applicable.

### 3.6 Changing options later

**\microtypesetup** {*<key = value list>*}

Inside the preamble, this command accepts all package options described above (except for **config**). In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts all options from section 3.1: **expansion**, **protrusion** and **activate**, which in turn may receive the values `true`, `false`, `compatibility` or `nocompatibility`, and **tracking**,  **Kerning** and **spacing** with the admissible values `true` or `false`. Passing the name of a font set is not allowed. Using this command, you could for instance temporarily disable font expansion by saying:

```
\microtypesetup{expansion=false}
```

## 4 Selecting fonts for micro-typography

By default, character protrusion will be applied to all text fonts used in the document, and a basic set of fonts will be subject to font expansion. You may want to customise which fonts should get the benefit of micro-typographic treatment. This can be achieved by declaring and activating ‘font sets’; these font sets are specified via font attributes that have to match.

`\DeclareMicrotypeSet` [*features*] {*set name*} {*set of fonts*}

`\DeclareMicrotypeSet*` This command declares a new set of fonts to which the micro-typographic extensions should be applied. The optional argument may contain a comma-separated list of features to which this set should be restricted. The starred version of the command declares *and* activates the font set at the same time.

The *set of fonts* is specified by assigning values to the NFSS font attributes: encoding, family, series, shape and size (cf. [L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> font selection](#)). Let’s start with an example. In the main configuration file `microtype.cfg`, a font set called ‘`basictext`’ is defined as follows:

```
\DeclareMicrotypeSet{basictext}
{ encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
  family   = {rm*,sf*},
  series    = {md*},
  size      = {normalsize,footnotesize,small,large}
}
```

If you now call

```
\UseMicrotypeSet[protrusion]{basictext}
```

in the document’s preamble, only fonts in the text encodings, roman or sans serif families, normal (or ‘medium’) series, and in sizes called by `\normalsize`, `\footnotesize`, `\small` or `\large`, will be protruded. Math fonts, on the other hand, will not, since they are in another encoding. Neither will fonts in bold face, or huge fonts. Etc.

If an attribute list is empty or missing – like the ‘shape’ attribute in the above example – it does not constitute a restriction. In other words, this is equivalent to specifying *all* possible values for that attribute. Therefore, the predefined set ‘`alltext`’, which is declared as:

```
\DeclareMicrotypeSet{alltext}
{ encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
```

is far less restrictive. The only condition here is that the encoding must match.

If a value is followed by an asterisk (like ‘`rm*`’ and ‘`sf*`’ in the first example), it does not designate an NFSS code, but will be translated into the document’s `\<value>default`, e.g., `\rmdefault`.<sup>4</sup> A single asterisk means `\<attribute>default`, e.g., `\encodingdefault`, respectively `\normalsize` for the size axis. Sizes may either be specified as a dimension (‘10’ or ‘10pt’), or as a size selection command *without* the backslash. You may also specify ranges (e.g., ‘`small-Large`’); while the lower

4 These translations will take place `\AtBeginDocument`, which means that changes to the defaults inside the preamble will also be taken into account. Only in cases where you change font defaults `\AtBeginDocument` yourself, you need to load `microtype` after these changes.

Table 2:

Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
all	Ø	Ø	Ø	Ø	Ø
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	Ø	Ø	Ø	Ø
alltext-nott (allmath-nott)	Text encodings, TS1 (OML, OMS, U)	\rm*, \sf*	Ø	Ø	Ø
basictext (basicmath)	Text encodings (OML, OMS)	\rm*, \sf*	\md*	Ø	\normalsize, \footnotesize, \small, \large
smallcaps	Text encodings	Ø	Ø	\sc*,si,scit	Ø
footnotesize	Text encodings, TS1	Ø	Ø	Ø	-\small
scriptsize	Text encodings, TS1	Ø	Ø	Ø	-\footnotesize
normalfont	\encoding*	\family*	\series*	\shape*	\normalsize
'Text encodings' = OT1, T1, T2A, LY1, OT4, QX, T5, EU1, EU2, TU      '\...*' = '\...default'					

boundary is included in the range, the upper boundary is not. Thus, '12-16' would match 12 pt, 13.5 pt and 15.999 pt, for example, but not 16 pt. You are allowed to omit the lower or upper bound ('-10', 'large-').

Additionally to this declaration scheme, you can add single fonts to a set using the 'font' key, which expects the concatenation of all font attributes, separated by forward slashes, i.e., 'font = <encoding>/<family>/<series>/<shape>/<size>'. This allows you to add fonts to the set that are otherwise disjunct from it. For instance, if you wanted to have the roman family in all sizes protruded, but only the normal sized, possibly italic, typewriter font (in contrast to, say, the small one), this is how you could declare the set:

```
\DeclareMicrotypeSet[protrusion]
{ myset }
{ encoding = T1,
  family   = rm*,
  font      = {T1/tt*/m/n/*,
               T1/tt*/m/it/*} }
```

As you can tell from the example, the asterisk notation is also permitted for the font key. A single asterisk is equivalent to '\*/\*/\*/\*/\*', i.e., the normal font. Size selection commands are possible, too, however, ranges are not allowed.

Table 2 lists the eleven predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, kerning and spacing when loading the package, for example:

```
\usepackage[protrusion=allmath,tracking=smallcaps]{microtype}
```

`\UseMicrotypeSet` [*features*] {*set name*}

This command activates a font set previously declared by `\DeclareMicrotypeSet`. Using the optional argument, you can limit the application of the set to one or more features. This command only has an effect if the feature was activated in the package options.

`\DeclareMicrotypeSetDefault` [*features*] {*set name*}

If a feature is enabled but no font set has been chosen explicitly, the sets declared by this command will be activated. By default, the ‘alltext’ font set will be activated for character protrusion and additional kerning, the ‘alltext-nott’ set for font expansion and interword spacing, and the ‘smallcaps’ set for tracking.

These commands may only be used in the preamble or in the main configuration file. Their scope is global to the document. Only one set per feature may be activated.

## 5 Micro fine tuning

Every character asks for a particular protrusion, kerning or spacing amount. It may also be desirable to restrict the maximum expansion of certain characters. Furthermore, since every font looks different, settings have to be specific to a font or set of fonts. This package offers flexible and straight-forward methods of customising these finer aspects of micro-typography.

All fine-tuning commands follow basically the same syntax: they all take three arguments; the first one is optional and may contain additional options; in the second argument, you specify the set of fonts to which the settings should apply; the third argument contains the actual settings. Here, as in all configuration commands, all spaces are ignored.

The set of fonts to which the settings should apply is declared using the same syntax of *font axis* = *value list* pairs as for the command `\DeclareMicrotypeSet` (see section 4), with the only difference that values including asterisks (which, as you may recall, stand for the respective default) will be translated immediately instead of at the end of the preamble. To find the matching settings for a given font the package will try all combinations of font encoding, family, series, shape and size, with decreasing significance in this order. For instance, if settings exist for both the current family (say, T1/cmr//) and for italic fonts in the normal weight (T1/m/it/), the settings for the cmr family would apply. The encoding must always match.

The characters may be specified either as a single letter (A), as a text symbol command (`\textquoteleft`), or as a slot number (resp. Unicode number for LuaTeX or XeTeX): three or more digits for decimal notation, prefixed with “ for hexadecimal, with ‘ for octal numerals (e.g., the ‘fl’ ligature in T1 encoding: 029, “1D, ‘35). 8-bit (and even UTF-8) characters may be entered directly or in L<sup>A</sup>T<sub>E</sub>X’s traditional 7-bit notation: both “A and Ä are valid, provided the character is actually declared in both the input and the font encoding. With LuaTeX or XeTeX, you may additionally specify a (font-specific) glyph name, prefixed with ‘/’ (e.g., the ‘fl’ ligature as /f\_l). Note that you also have the possibility to declare lists of characters that should inherit settings (see section 5.6).

## 5.1 Character protrusion

pdfTeX 0.14f | LuaTeX 0.30 | XeTeX 0.9997

**\SetProtrusion** [*options*] {*set of fonts*} {*protrusion settings*}

Using this command, you can set the protrusion factors for each character of a font or a set of fonts. A very incomplete example would be the following:

```
\SetProtrusion
{ encoding = T1,
  family   = cmr }
{ A        = {50,50},
  \textquoteleft = {700, } }
```

which would result in the character ‘A’ being protruded by 5% of its width on both sides, and the left quote character by 70% of its width into the left margin. This would apply to all font shapes, series and sizes of the T1 encoded Computer Modern Roman family.

The *protrusion settings* consist of *character* = *protrusion factors* pairs. The protrusion factors designate the amount that a character should be protruded into the left margin (first value) respectively into the right margin (second value). By default, the values are relative to the character widths, so that a value of 1000 means that the character should be shifted fully into the margin, while, for example, with a value of 50 it would be protruded by 5% of its width. Negative values are admitted, as well as numbers larger than 1000 (but effectively not more than 1em of the font). You may omit either number if the character should not be protruded on that side, but must not drop the separating comma.

*Options:*

**name** You may assign a name to the protrusion settings, so that you are able to load it by another list.

**load** You can load another list (provided, you assigned a name to it) before the current list will be loaded, so that the fonts will inherit the values from the loaded list.

In this way, the configuration may be simplified considerably. You can for instance create a default list for a font; settings for other shapes or series can then load these settings, and extend or overwrite them (since the value that comes last will take precedence). Font settings will be loaded recursively. The following options will affect all loaded lists, in other words, any options from the loaded lists will be ignored:

**factor** This option can be used to influence all protrusion factors of the list, overriding any global factor setting (see section 3.2). For instance, if you want fonts in larger sizes to be protruded less, you could load the normal lists, just with a different factor applied to them:

```
\SetProtrusion
[ factor = 700,
  load   = cmr-T1 ]
{ encoding = T1,
  family   = cmr,
  size     = large- }
{ }
```

**unit** By default, the protrusion factors are relative to the respective character's width. The `unit` option may be used to override this and make `microtype` regard all values in the list as thousandths of the specified width. Issuing, for instance, `'unit=1em'` would have the effect that a value of, say, 50 now results in the character being protruded by 5% of an em of the font (thus simulating the internal measuring of pdfTeX's `\lcode` and `\rcode` primitives). The default behaviour can be restored with `unit=character`.<sup>5</sup>

**preset** Presets the protrusion codes of all characters to the specified values (`={\langle left \rangle, \langle right \rangle}`), possibly scaled by a factor. A `unit` setting will only be taken into account if it is not `=character`.

**inputenc** Selects an input encoding that should apply to this list, regardless of what the document's input encoding is. You may specify any encoding that can be loaded via the `inputenc` package, e.g., `ansinew`, `koi8-r`, `utf8`.

**context** The scope of the list may be limited to a certain context. For further details, see section 6.

## 5.2 Font expansion

pdfTeX 0.14f | LuaTeX 0.30

`\SetExpansion` [*options*] {*set of fonts*} {*expansion settings*}

By default, all characters of a font are allowed to be stretched or shrunk by the same amount. However, it is also possible to limit the expansion of certain characters if they are more sensitive to deformation. This is the purpose of the `\SetExpansion` command. Note that it will only have an effect if the package has been loaded with the selected option (cf. section 3.3). Otherwise, the expansion settings will be ignored – unlike the options in the optional first argument, which will still be evaluated. If the selected option has been set to true, and settings for a font don't exist, font expansion will not be applied to this font at all. Should the extraordinary situation arise that you want to employ selected expansion in general but for a particular font (set) all characters should be expanded or shrunk by the same amount, you would have to declare an empty list for these fonts.

The *expansion settings* consist of *character* = *expansion factor* pairs. You may specify one number for each character, which determines the amount that a character may be expanded. The numbers denominate thousandths of the full expansion. For example, if you set the expansion factor for the character 'O' to 500, it will only be expanded or shrunk by one half of the amount that the rest of the characters will be expanded or shrunk. While the default value for character protrusion is 0 – that is, if you didn't specify any characters, none would be protruded – the default value for expansion is 1000, which means that all characters would be expanded by the same amount.

*Options:*

**name, load, preset, inputenc, context** Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset

<sup>5</sup> The `unit` option can even be passed globally to the package (cf. section 3.2). However, all provided settings are created under the assumption that the values are relative to the character width. Therefore, you should only change it if you are certain that the default settings will not be used in your document.

all expansion factors, to set the input encoding, or to determine the context of the list (expansion contexts are only possible with pdfTeX version 1.40.4 or newer).

**auto**, **stretch**, **shrink**, **step** These keys can be used to override the global settings from the package options (see section 3.3). If you don't specify either one of stretch, shrink and step, their respective global value will be used (that is, no calculation will take place).

As a practical example, suppose you have a paragraph containing a widow that could be avoided by shrinking the font a bit more. In conjunction with the context option (see section 6 for further details), you could thus allow for more expansion in this particular paragraph:

```
\SetExpansion
[ context = sloppy,
  stretch = 30,
  shrink   = 60,
  step     = 5 ]
{ encoding = {OT1,T1,TS1} }
{ }
% ... END PREAMBLE
{\microtypecontext{expansion=sloppy}%
This paragraph contains a `fussy' widow.}
```

This method of employing contexts to temporarily apply different expansion parameters only works with pdfTeX version 1.40.4 or later,<sup>6</sup> or with LuaTeX. Also note that both pdfTeX and LuaTeX prohibit the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the `sloppy` context would have to be applied to complete paragraphs.

**factor** This option provides a different method to alter expansion settings for certain fonts, working around the restriction just mentioned. The `factor` option influences the expansion factors of all characters (in contrast to the overall stretchability) of the font. For instance, if you want the italic shape to be expanded less, you could declare:

```
\SetExpansion
[ factor = 500 ]
{ encoding = *,
  shape    = it }
{ }
```

The `factor` option can only be used to *decrease* the stretchability of the characters, that is, it may only receive values smaller than 1000. Also, it can only be used for single fonts or font sets; setting it globally in the package options wouldn't make much sense – to this end, you use the package's `stretch` and `shrink` options.

### 5.3 Tracking

pdfTeX 1.40 | LuaTeX 0.62

`\SetTracking` [*options*] {*set of fonts*} {*tracking amount*}

An important typographic technique – which was missing in TeX for a long time – is the adjustment of tracking, i.e., the uniform addition or subtraction of letter space

<sup>6</sup> For older versions, a dirty trick is laid out in section 14.2 on page 59.

to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.<sup>7</sup> The `\SetTracking` command allows specifying the tracking amount for different fonts or font sets. It will also be evaluated by the `\textls` command, which may be used for letterspacing shorter pieces of text (see section 7).

The *tracking amount* is specified in thousandths of 1em (or the given unit); negative values are allowed, too.

*Options:*

**name, unit, context** These options serve the same functions as in the previous configuration commands. The unit may be any dimension, default is 1em.

**spacing** When the inter-*letter* spacing is altered, the inter-*word* spacing probably also needs to be adjusted. This option expects three numbers for interword space, stretch and shrink respectively, which are given in thousandths of 1em (or of the current unit). If a value is followed by an asterisk, it denotes thousandths of the respective font dimension which will be added to it. For instance, with

```
\SetTracking[ spacing = {25*,166, } ]{ encoding = *, shape = sc }{ 25 }
```

the interword space will be increased by 2.5%, the stretch amount will be set to 0.166em, while the shrink amount will be left untouched. If you don't specify the spacing option, the interword space will be scaled by the current letterspace amount (as in the above example), while stretch and shrink will not be changed.

**outer spacing** If an interword space immediately precedes or follows letter-spaced text, it will by default be equal to that within the text. With this option, which accepts the same values as **spacing**, it may be adjusted independently.

**outer kerning** If, on the other hand, no interword space precedes or follows, you may still want to slightly set off the first and last letter from adjoining letters. This option expects the kerning amounts for left and right hand side, separated by a comma, in thousandths of 1em (or the current unit). If a value is followed by an asterisk, it denotes thousandths of the current letterspacing amount. A single asterisk means '500\*'; this is also the default, i.e., the sum of the outer kerns is by default equal to the current letterspace amount. To remove kerning on both sides, you would write 'outer kerning={0,0}'.

**no ligatures** By default, ligatures in letterspaced fonts will be constructed as usual, which may be advisable when changing the tracking by only a small amount. For larger letterspacing amounts, on the other hand, the normal letter space within ligatures would have displeasing effects. This key expects a comma-separated list of characters for which ligatures should be disabled; only the character that begins a ligature must be specified. If the key is given without a value, *all* ligatures of the font will be disabled. With pdfTeX, this is not recommended, however, since it entails that kerning will be switched off, too. With LuaTeX, there is no such limitation. The default settings disable ligatures for the character 'f' only, i.e., 'ff',

<sup>7</sup> With full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.



‘fi’, ‘ffi’, etc.<sup>8</sup> In exceptional situations, you can manually break up a ligature by inserting ‘{\kern0pt}’ resp. babel’s “|” shortcut, or protect it by enclosing it in \slig (see section 7).

Since a picture is worth a thousand words, probably even more if, in our case, it depicts a couple of letterspaced words, let’s bring one to sum up these somewhat confusing options. Suppose you had the following settings (which are in no way recommended; they only serve illustrative purposes):

```
\SetTracking
[ no ligatures = {f},
  spacing      = {600*,-100*, },
  outer spacing = {450,250,150},
  outer kerning = {*,*} ]
{ encoding = * }
{ 160 }
```

and then write:

```
Stop \textls{stealing sheep}!
```

this would be the (typographically dubious) outcome:

Stop stealing sheep!

While the word ‘Stop’ is not letterspaced, the space between the letters in the other two words is expanded by the *tracking amount* of  $160/1000\text{em} = 0.16\text{em}$ . The *inner space* within the letterspaced text is increased by 60%, while its *stretch* amount is decreased by 10% and the *shrink* amount is left untouched. The *outer space* (of 0.45em) immediately before the piece of text may *stretch* by 0.25em and *shrink* by 0.15em. Note that there is no outer space after the text, since the exclamation mark immediately follows; instead, the default *outer kern* of half the letterspace amount (0.08em) is added. Furthermore, one *ligature* wasn’t broken up, because we neglected to specify the ‘s’ in the `no ligatures` key.

As another, more realistic example, suppose you want to space out all small capitals by  $50/1000\text{em}$ , fonts smaller than \small by 0.02em, and to decrease the tracking of large type by 0.02em. This could be achieved with the following settings:

```
\usepackage[tracking=true]{microtype}
\DeclareMicrotypeSet*[tracking]{my}
{ encoding = *,
  size      = {-small,Large-},
  font      = */*/*/sc/* }
\SetTracking[ no ligatures = f ]{ encoding = *, shape = sc}{ 50 }
\SetTracking{ encoding = *, size = -small }{ 20 }
\SetTracking{ encoding = *, size = Large- }{ -20 }
```

Letterspaced fonts for which settings don’t exist will be spaced out by the default of 0.1em (adjustable with the package option `letterspace`, see section 3.5). Suppose

<sup>8</sup> With pdfTeX versions older than 1.40.4, *all* ligatures, and hence all kerning, will be disabled. It is therefore recommended to use at least version 1.40.4.

Click on the image to show the kerns and spacings involved. Click on emphasised words in the text below to reveal the relation of image and code.

your editor wants you to shorten your 1000-pages chef-d'œuvre by a handful of pages, you could load microtype with (fingers crossed):

```
\usepackage[tracking=alltext,letterspace=-40]{microtype}
```

## 5.4 Additional kerning

pdfTeX 1.40

`\SetExtraKerning` [*options*] {*set of fonts*} {*kerning settings*}

With this command, you can fine tune the extra kerning. In contrast to standard kerning, which is always associated with a *pair* of characters, and to tracking, which specifies the space between *all* characters of a font, the extra kerning relates to single characters, that is, whenever a particular character appears in the text, the specified kerning will be inserted, regardless of which character precedes resp. follows it. (Put differently, this feature allows modifying the left or right *sidebearings* of specific glyphs.)

It should not be neglected to mention a limitation of this feature: words *immediately following* such a kern (not separated by a space) will not be hyphenated, unless you insert the breakpoints manually, e.g., for kerning after the apostrophe, 'l'apostrophe'. Furthermore, additional kerning will not be applied in math mode. These restrictions of pdfTeX will hopefully be lifted some time.

The *kerning settings* are specified as pairs of *character* = *kerning values*, where the latter consist of two values: the kerning added before the character, and the kerning appended after the respective character. Once again, either value may be omitted, but not the separating comma.

*Options:*

**name**, **load**, **factor**, **preset**, **inputenc** These options serve the same function as in the previous configuration commands.

**unit** Admissible values are: space, character and a *dimension*. By default, the values denote thousandths of 1 em.

**context** When it comes to kerning settings, this option is especially useful, since it allows applying settings depending on the current language.

For example, you can find the following settings, intended to be used for documents written in French, in the main configuration file:

```
\SetExtraKerning
[ name      = french-default,
  context    = french,
  unit       = space ]
{ encoding = {OT1,T1,LY1} }
{
  : = {1000,}, % = \fontdimen2
  ; = {500, }, % ~ \thinspace
  ! = {500, },
  ? = {500, }
}
```

What is the result of these settings? If they are active, like in the current paragraph, a thin space will be inserted in front of each question mark, exclamation mark and

semicolon; a normal space in front of the colon. Read section 6 to learn how to activate these settings! This paragraph was input like this :

```
\begin{microtypecontext}{\kerning=french}
What is the result of these settings? If they are active, like in the
current paragraph, a thin space will be inserted in front of each
question mark, exclamation mark and semicolon; a normal space in front
of the colon. Read section~\ref{sec:context} to learn how to activate
these settings! This paragraph was input like this:
\end{microtypecontext}
```

## 5.5 Interword spacing

pdfTeX 1.40

`\SetExtraSpacing` [*options*] {*set of fonts*} {*spacing settings*}

This command allows you to fine tune the interword spacing (also known as glue). A preliminary remark on what a ‘space’ is may be in order: between two words, T<sub>E</sub>X will insert a so called glue, which is characterised by three parameters – the normal distance between two words, the maximum amount of space that may be added to it, and the maximum amount that may be subtracted. The latter two parameters come into effect whenever T<sub>E</sub>X tries to break a paragraph into lines and does not succeed; it can then stretch or shrink the spaces between words. These three parameters are specific to each font.

On top of these glue dimensions, T<sub>E</sub>X has the concept of ‘space factors’. They may be used to increase the space after certain characters, most prominently the punctuation characters. pdfT<sub>E</sub>X’s additional spacing adjustment may be considered as an extension to space factors with much finer control: while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – you may modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. Note that when interword spacing adjustment is in effect, space factors are ignored.

The *spacing settings* are declared as pairs of *character* = *spacing factors*, where the latter consist of three numbers: first, the additional kern inserted after this character if it appears before an interword space, second, the additional stretch amount, and third, the additional shrink amount. All values may also be negative, in which case the dimensions will be decreased. Not all values have to be specified, but the settings must always contain the two separating commas.

*Options:*

**name**, **load**, **factor**, **preset**, **inputenc**, **context** These options serve the same function as in the previous configuration commands.

**unit** You can specify the unit by which the specified numbers are measured. Possible values are: *character*, a *dimension* and, additionally, *space*. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with the following (nonsensical) settings:

```
\SetExtraSpacing
[ unit = space ] % default
{ font = */*/*/*/* }
{
  . = {1000,1000,1000},
}
```

the space inserted after a full stop would be doubled (technically speaking:  $2 \times \text{\fontdimen 2}$ ), as would the maximum stretch and shrink amounts of the interword space ( $\text{\fontdimen 3}$  and  $\text{\fontdimen 4}$ ). Conversely, setting all three values to  $-1000$  would completely cancel a space after the respective character.

## 5.6 Character inheritance

`\DeclareCharacterInheritance` [*features*] {*set of fonts*} {*inheritance lists*}

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters À, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list of features, you can confine the scope of the list. Additionally, it accepts the `inputenc` key to set the input encoding for this list. The font set can be declared in the usual way. The inheritance lists are declared as pairs of *base character* = *list of inheriting characters*. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

The situation is different with LuaTeX and XeTeX, however: the default inheritance settings only contain those glyphs that can safely be assumed to exist in any font; but since OpenType fonts may contain many more glyphs for different scripts (languages), it is quite probable that font-specific settings are necessary, which should be specified in the font's configuration file (see next section).

## 5.7 Configuration files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion, expansion, spacing and kerning settings, will be loaded from the file `microtype.cfg`. You may extend this file with custom settings (or load a different configuration file with the 'config' option, see section 3.5).

If you embark on creating new settings for a font family, you should put them into a separate file, whose name must be: `mt-font family.cfg` (e.g., `mt-cmr.cfg`; any spaces in the font name should be removed, e.g., `mt-MinionPro.cfg`), and may contain all commands described in the current section 5. These files will be loaded automatically if you are actually using the respective fonts. This package ships with configuration files for a number of font families. Table 3 lists them all.

`\DeclareMicrotypeVariants` {*list of suffixes*}

`\DeclareMicrotypeVariants*` On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a 'variant' of the base font (cf. Karl Berry's [Fontname](#)). It is thus possible to put settings for, e.g., the fonts `pplx` (expert set), `pplj` (oldstyle numerals) and `ppl` (plain) into one and the

Table 3:

Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
<i>Type 1 fonts</i>	Encodings	Shapes
Generic	OT1, T1, T2A, LY1, QX, (TS1) <sup>a</sup>	n, (it, sl, sc) <sup>a</sup>
Computer Modern Roman (cmr) <sup>b</sup>	OT1, OT4, T1, T2A, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (bch) <sup>c</sup>	OT1, T1, T5, LY1, TS1	n, it, (sl) <sup>d</sup> , sc
EB Garamond <sup>e</sup>	OT1, T1, LY1, TS1	n, it, (sl) <sup>d</sup> , sc
URW Garamond (ugm) <sup>f</sup>	OT1, T1, TS1	n, it
Bitstream Letter Gothic (blg) <sup>g</sup>	OT1, T1, TS1	n, it
Adobe Minion (pmnx, pmnj)	OT1, T1, T2A, LY1, TS1	n, it, (sl) <sup>d</sup> , sc, si
Palatino (ppl, pplx, pplj) <sup>h</sup>	OT1, OT4, T1, LY1, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Times (ptm, ptmx, ptmj) <sup>i</sup>	OT1, OT4, T1, LY1, QX, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Computer Modern math (cmsy, cmm) <sup>j</sup>	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) <sup>k</sup>	U	n
Euro symbols (Adobe, ITC, marvosym)	U/OT1	n, it
<i>OpenType fonts</i>	Scripts	Shapes
Generic	Latin	n, (it, sl, sc) <sup>a</sup>
Latin Modern Roman <sup>l</sup>	Latin, Greek	n, it, (sl) <sup>d</sup>
Charis SIL	Latin, Cyrillic, Greek	n, it, sc
Palatino <sup>m</sup>	Latin	n, it, sc

<sup>a</sup> Incomplete  
<sup>b</sup> Aliases: Latin Modern Roman (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor), mmodern (mlmr)  
<sup>c</sup> Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr), XCharter  
<sup>d</sup> Settings inherited from italic shape  
<sup>e</sup> Alias: Adobe Garamond (pad, padx, padj)  
<sup>f</sup> Aliases: mathdesign/URW Garamond (mdugm), garamondx (zgm, zgmj)  
<sup>g</sup> Alias: ulgothic (ulg)  
<sup>h</sup> Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T<sub>E</sub>X Gyre Pagella (qpl), newpx, FPL Neu (fp9x, fp9j), domitian  
<sup>i</sup> Aliases: txfonts (txr), qfonts/QuasiTimes, T<sub>E</sub>X Gyre Termes (qtm), newtx, tempora, step, stix/stix2  
<sup>j</sup> Aliases: Latin Modern (lmsy, lmm), mmodern (mlmsy, mlmm)  
<sup>k</sup> Alias: eulervm (zeur, zeus)  
<sup>l</sup> Alias: New Computer Modern  
<sup>m</sup> Aliases: Palatino Linotype, Palatino LT Std, T<sub>E</sub>X Gyre Pagella, Domitian

same file `mt-ppl.cfg`. This command expects a comma-separated list of variant suffixes. The starred version appends the suffix(es) to the existing list. The default declaration in `microtype.cfg` is:

```
\DeclareMicrotypeVariants{x,j,w,a,d,0,1,-LF,-TLF,-OsF,-T0sF}
```

```
\DeclareMicrotypeAlias {<font name>} {<alias font>}
```

This command may be used for fonts that are very similar, or actually the same (for instance if you did not stick to the Berry naming scheme when installing a font). An example would be the Latin Modern fonts, which are derived from Computer Modern, so that it is not necessary to create new settings for them – you could say:

```
\DeclareMicrotypeAlias{lmr}{cmr}
```

which would make the package, whenever it encounters the font `lmr` and does not find settings for it, also try the font `cmr`. In fact, you will find this very line, along with some others, in the default configuration file.

```
\LoadMicrotypeFile {<font name>}
```

In rare cases, it might be necessary to load a font configuration file manually, for instance, from within another configuration file, or to be able to extend settings defined in a file that would otherwise not be loaded automatically, or would be loaded too late.<sup>9</sup> This command will load the file `'mt-<font name>.cfg'`.

## 6 Context-sensitive setup

The microtype package also allows applying different micro-typographic settings to the fonts depending on the context in which they occur. This opens up the space for infinite possibilities of tweaking the document's appearance.

```
\microtypecontext {<context assignments>}
```

This command may be used anywhere in the document (also in the preamble) to change the micro-typographic context in the current group. To each feature (**protrusion**, **expansion**, (or **activate** as a shortcut for both), **tracking**, **spacing** and **kerning**), one context may be assigned. Consequently, only settings with the corresponding 'context' keyword will be applied.

```
\begin{microtypecontext} {<context assignments>}
```

```
\end{microtypecontext}
```

Like many L<sup>A</sup>T<sub>E</sub>X commands, it is also available in the form of an environment.

```
\textmicrotypecontext {<context assignments>} {<general text>}
```

As another possibility, the command `\textmicrotypecontext` sets the context(s) for the text given in the second argument.

Suppose you want the footnote markers in the text to be protruded by a larger amount. You could define settings for the numbers:

```
\SetProtrusion
[ context = footnote ]
{ font = */*/*/scriptsize } % adapt if necessary
{ 1 = { ,650}, 2 = { ,400}, 3 = { ,400}, 4 = { ,400}, 5 = { ,400},
  6 = { ,400}, 7 = { ,500}, 8 = { ,400}, 9 = { ,400}, 0 = { ,400} }
```

and have the context changed in the footnote marker command. This command differs among the various classes; for the base classes, e.g., `article`, it would be:

```
\newcommand*\new@makefnmark{\hbox{\@textsuperscript{\normalfont
\microtypecontext{protrusion=footnote}\@thefnmark}}}
\renewcommand*\@footnotemark{%
\leavevmode \ifhmode\edef\x@sf{\the\spacefactor}\nobreak\fi
\new@makefnmark \ifhmode\spacefactor\x@sf\fi \relax}
```

<sup>9</sup> Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 14.4.4.

For the memoir class, you would additionally have to disable auto-detection of multiple footnotes, which prevents protrusion:

```
\renewcommand*\makefnmark{\hbox{\@textsuperscript{\normalfont
\microtypecontext{protrusion=footnote}\@thefnmark}}}
\let\m@mmf@prepare\relax
\let\m@mmf@check\relax
```

Another possibility would be to employ contexts for a language-dependent setup. For instance, if you are writing a text in French, you could add:

```
\microtypecontext{kerning=french}
```

to the preamble. This would have the effect that kerning settings for the French context would be applied to the document. Should parts of the document be in English, you could write:

```
\textmicrotypecontext{kerning=}{English text!}
```

to reset the context, so that the punctuation characters in these parts will not receive any extra kerning.

Instead of adding these commands manually to your document, you may also load microtype with the babel option (see section 3.5). The current language will then be automatically detected and the contexts set accordingly.

`\DeclareMicrotypeBabelHook`  $\{\langle list\ of\ babel\ languages \rangle\} \{\langle context\ list \rangle\}$

Naturally, microtype does not know about the typographic specialties of every language. This command is a means of teaching it how to adjust the context when a particular language is selected. The main configuration file contains among others the following declaration:

```
\DeclareMicrotypeBabelHook
{french,français,acadian,canadien}
{kerning=french, spacing=}
```

Consequently, whenever you switch to the French language, the kerning context will be changed to ‘french’ and the spacing context will be reset. This hook only has an effect if the package was loaded with the babel option. Currently, microtype supports French and Turkish kerning and English spacing (aka. `\nonfrenchspacing`). For unknown languages, all contexts will be reset.

## 7 Letterspacing revisited

pdfTeX 1.40 | LuaTeX 0.62

`\textls`  $[\langle amount \rangle] \{\langle general\ text \rangle\}$

While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.<sup>10</sup> For such ad-hoc letterspacing, microtype introduces two commands that can be used (independently of whether the tracking option is enabled) in the same way as L<sup>A</sup>T<sub>E</sub>X’s text commands: `\textls` – which also works in math mode – expects the text in the mandatory argument, while `\lsstyle` will

`\lsstyle`

<sup>10</sup> Letterspacing should be used cautiously; in particular, letterspacing lowercase text is held in abhorrence by honourable typographers. Unless you know what you are doing, you should probably only letterspace capitals or small capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.

switch on letterspacing for all subsequent fonts until the end of the current group.

`\textls*` The starred version of `\textls` does not add any extra kerning before or after the text, which may be useful, e.g., for section titles. By default, each character will be spaced out by  $100/1000\text{em} = 0.1\text{em}$ ; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

`\lslig` `{\langle ligature \rangle}`

Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you need not worry about protecting or breaking ligatures with most fonts. However, in certain situations, there may be a conflict of ligatures beginning with the same letter, where some of them should be inhibited, while others should not. When letterspacing text typeset in Fraktur fonts, for example, the ligatures ‘ch’, ‘ck’, ‘tz’ and ‘sz’ (‘ß’) should never be broken up; you also usually see the ‘st’ (‘t’) ligature in letterspaced text. Furthermore, at least the `yfonts` package realises the short s (‘s’) as the ligature ‘s:’. On the other hand, the ‘ct’ ligature and the other ‘long s’ ligatures often found in Fraktur fonts should be suppressed. There are two ways of solving this problem: either don’t disable the ‘s’ and/or ‘c’ ligatures and break those that need to be broken up by inserting ‘`\kern0pt`’ or babel’s “|” shortcut; or disable them and protect those ligatures that need to be protected by enclosing them in the `\lslig` command. So, the following two solutions have the same result (namely, ‘Ausſichtslosigkeit’, with ligatures shown in green, inhibited ligatures in red).

```
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{120}
\textfrak{\lsstyle Aus{s{\kern0pt}ichts:los{\kern0pt}igkeit}
```

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{120}
\textfrak{\lsstyle Au\lslig{s:}si\lslig{ch}t\lslig{s:}losigkeit}
```

`letterspace.sty` These three commands (plus the `letterspace` option, described in section 3.4) are also available with the alternative `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions (and also omitting the possibilities of the `\SetTracking` command – all ‘f’ ligatures will be disabled, inner and outer spacing and outer kerning will be set to the default values described in section 5.3). If you prefer to forgo `microtype`’s specialties, you may load the `letterspace` package instead. Both packages should not be used at the same time.

In contrast to `microtype`, which requires  $\text{\LaTeX}$ , the `letterspace` package also works with `eplain` or even only `miniltx`: for use with `eplain`, load the package with `\usepackage` inside the `\beginpackages ... \endpackages` environment; with `miniltx` (which does not support package options) simply `\input letterspace.sty`.

## 8 Disabling ligatures

pdfTeX 1.30 | LuaTeX 0.30

`\DisableLigatures` `[\langle characters \rangle] {\langle set of fonts \rangle}`

While completely disabling all ligatures of a font (which will also switch off kerning for this font), purposely *lowers* the micro-typographic quality instead of raising



it, it is especially useful for typewriter fonts, so that, e.g., in a T1 encoded font, `\texttt{--}` will indeed be printed as ‘--’, not as ‘-’. `\DisableLigatures` may be used to specify, in the usual way, a set of fonts for which ligatures should be disabled, for example, of the typewriter font in T1 encoding:

```
\DisableLigatures{encoding = T1, family = tt* }
```

It is also possible to disable selected ligatures only. The optional argument may contain a comma-separated list of characters for which the ligature mechanism should be inhibited:

```
\DisableLigatures[?,!]{encoding = T1} % inhibit ?‘ and !‘, but not fi, –, », etc.
```

Only the character that begins the ligature(s) should be specified. This command may only be used in the preamble, and only once.<sup>11</sup>

## 9 Hints and caveats

*Use settings that match your font.* Although the default settings should give reasonable results for most fonts, the particular font you happen to be using may have different character shapes that necessitate more or less protrusion. In particular, italic letter shapes may differ wildly in different fonts, hence I have decided against providing default protrusion settings for them. The file `test-microtype.tex` might be of some help when adjusting the protrusion settings for a font.

*Don’t use too large a value for expansion.* Font expansion is a feature that is supposed to enhance the typographic quality of your document by producing a more uniform greyness of the text block (and potentially reducing the number of necessary hyphenations). When expanding or shrinking a font too much, the effect will be turned into the opposite. Expanding the fonts by more than 2%, i.e., setting a stretch limit of more than 20, should be justified by a typographically trained eye. If you are so lucky as to be in the possession of multiple instances of a Multiple Master font, you may set expansion limits to up to 4%.

*Don’t use font expansion for web documents (with older pdfTeX versions).* With pdfTeX versions older than 1.40, each expanded instance of the font will be embedded in the PDF file, hence the file size may increase by quite a large factor (depending on expansion limits and step). Therefore, courtesy and thriftiness of bandwidth command it not to enable font expansion when creating files to be distributed electronically. With pdfTeX 1.40 and LuaTeX, which use a different technique of expansion, the increase of file size can be neglected.

*You might want to disable protrusion in the Table of Contents.* In unfortunate situations, enabled protrusion might internally alter the line length in the TOC and similar lists in such a way that an excess leader dot will fit in. The solution is to temporarily disable protrusion for the TOC:

```
\microtypesetup{protrusion=false}
\tableofcontents
\microtypesetup{protrusion=true}
```

<sup>11</sup> With LuaTeX, you have to load the fonts with the `fontspec` option ‘`Renderer=Basic`’.

*You might want to disable protrusion in verbatim environments.* As you know by now, `microtype` will by default activate character protrusion for all fonts contained in the font set ‘alltext’. This also includes the typewriter font. Although it does make sense to protrude the typewriter font if it appears in running text (like, for example, in this manual), this is probably not desirable inside the `verbatim` environment. However, `microtype` has no knowledge about the context that a font appears in but will solely decide by examining its attributes. Therefore, you have to take care of disabling protrusion in `verbatim` environments for yourself (that is, if you don’t want to disable protrusion for the typewriter font altogether, by activating, say, the font set ‘alltext-nott’). While the `\microtypesetup` command has of course been designed for cases like this, you may find it tiresome to repeat it every time if you are using the `verbatim` environment frequently. The following line (which requires the `etoolbox` package), added to the document’s preamble, would serve the same purpose:

```
\AtBeginEnvironment{verbatim}{\microtypesetup{activate=false}}
```

If you are using the `fancyvrb` or the `listings` package, this is not necessary, since their implementation of the corresponding environments will inhibit protrusion anyway.

*Settings for Greek/Thai/Armenian etc. encodings are not yet included.* The default sets of fonts for which the micro-typographic features will be enabled (see table 2) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e.g., LGR, T2B, etc.), `microtype` will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion at least, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

*Only employ kerning adjustment if it is customary in the language’s typographic tradition.* In contrast to protrusion and expansion, additional kerning does not unconditionally improve the micro-typographical quality of your document. You should only switch it on if you are writing a document in a language whose typographic tradition warrants such kerning. If you are, for example, writing an English text, your readers would probably be rather confused by additional spaces before the punctuation characters.

*Adjustment of interword spacing is still experimental.* The implementation of this feature in `pdfTeX` is not complete, and may not yield the positive effects on the typographical quality you might expect – in certain situations, there may even be undesired side effects, in particular, when used together with the `ragged2e` package. Therefore, the `spacing` option should not be chosen blindly; it is also recommended to experiment with the settings in order to understand the workings of this feature.

*Compatibility and interaction with other packages:* The `microtype` package is supposed to work happily together with all other `LATEX` packages (except for `pdfcpot`). However, life isn’t perfect, so problems are to be expected. Currently, I am aware of the following issues:

- Even though all configuration files are still provided in legacy (7-bit) format, using multi-byte (Unicode) characters in the settings should run smoothly with an up-to-

date L<sup>A</sup>T<sub>E</sub>X system. For older systems or documents in legacy encodings, in contrast, this requires loading the `inputenc` package first. Furthermore, when using multiple input encodings in a document, 8-bit characters in the settings will only work reliably if you specify the `inputenc` key.

- When loading the package with the `babel` option, you must load the `babel` package before `microtype`.
- Before this package was fully compatible with Lua<sub>T</sub><sub>E</sub>X, the following method of enabling expansion and protrusion with the `fontspec` package was most often found to be recommended:

```
\newfontfeature{Microtype}{protrusion=default;expansion=default}
\defaultfontfeatures{Microtype}
```

This code should *not* be used with this package, as it will basically override all of the settings made by `microtype` – despite the naming, the above lines have nothing to do with this package.<sup>12</sup>

- With pdf<sub>T</sub><sub>E</sub>X, it is currently not possible to create character-specific settings for Chinese/Japanese/Korean fonts. Therefore, the only micro-typographic extension that can be made to work with CJK fonts is (non-selected) font expansion.
- When used with the `xeCJK` package or the `luatexja` package, text commands (e.g., `\A`, `\textless`) in the configuration will not be understood. You therefore have to ensure that `microtype` will encounter none of them. This requires, firstly, that the glyphs be specified only as single (possibly Unicode) characters, as numbers, or as glyph names (cf. section 5); and secondly, if you are using a font for which pre-defined settings do not exist, that you create these settings yourself (because otherwise, the default settings will be loaded, which do contain text commands). Furthermore, you should load `microtype` late.

*Possible error messages and how to get rid of them (specs may differ):*

- ! Font csnameendcsname=*cmr10+20 at 10.0pt* not loadable: Metric (TFM) file not found.  
This error message will occur if you are trying to employ font expansion while creating DVI output. Remember that *automatic* font expansion only works when running pdf<sub>T</sub><sub>E</sub>X or Lua<sub>T</sub><sub>E</sub>X in PDF mode. Although expansion is also possible in DVI mode with pdf<sub>T</sub><sub>E</sub>X, it requires that all instances of the expanded fonts exist on your <sub>T</sub><sub>E</sub>X system.
- ! pdf<sub>T</sub><sub>E</sub>X error (font expansion): auto expansion is only possible with scalable fonts.  
Automatic font expansion has been improved in pdf<sub>T</sub><sub>E</sub>X 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn't found at all, e.g., because of missing map entries.
- Warning: pdf<sub>L</sub><sub>A</sub>T<sub>E</sub>X: font *ptmr8r* cannot be expanded (not an included Type1 font)  
and the PDF viewer complains about a missing font, e.g., Adobe Reader thusly:  
Could not find a font in the Resources dictionary - using Helvetica instead.  
With pdf<sub>T</sub><sub>E</sub>X versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your

<sup>12</sup> They make use of features provided by `luaotfload` (via `fontspec`).

T<sub>E</sub>X system is not set up to embed (or ‘download’) the base PostScript fonts (e.g., Times, Helvetica, Courier). In most T<sub>E</sub>X distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to true.

- Warning: pdf<sub>l</sub>atex (file `ecrm1000+20`): Font `ecrm1000+20 at 1200` not found  
Furthermore, pdfT<sub>E</sub>X versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdfT<sub>E</sub>X versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font `T1/cmr/m/n/10=ecrm1000 at 10.0pt` not loaded: Not enough room left.  
Memory parameter ‘font\_mem\_size’ too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font\_max)=2000].  
Memory parameter ‘font\_max’ too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf\_mem\_size)=65536].  
Memory parameter ‘pdf\_mem\_size’ too small (pdfT<sub>E</sub>X versions older than 1.30).  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdfT<sub>E</sub>X may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, e.g., T<sub>E</sub>X Live, change the settings in `texmf.cnf`, for MiK<sub>T</sub>E<sub>X</sub>, in the file `miktex.ini` (2.4 or older) resp. `pdflatex.ini` (2.5 or newer).
- pdfT<sub>E</sub>X warning (font expansion): font should be expanded before its first use  
This warning will occur with pdfT<sub>E</sub>X versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

*The source code of this document is freely available.* If you wonder how this document was created, just have a look at the source code in `microtype.dtx`, which is either already included in your T<sub>E</sub>X distribution, or else can be downloaded from [CTAN](#). For the source code of the logo on the title page and of the letterspacing sample from section 5.3, see the appendices A and B. If you want to re-typeset the documentation, read the comments at the end of `microtype.dtx`.

## 10 Contributions

I would be glad to include configuration files for more fonts. Preparing such configurations is quite a time-consuming task and requires a lot of patience. To alleviate this process, this package also includes a test file that can be used to check at least the protrusion settings (`test-microtype.tex`). If you have created a configuration file for another font, or if you have any suggestions for enhancements in the default configuration files, I would gratefully accept them: [w.m.l@gmx.net](mailto:w.m.l@gmx.net).

## 11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn’t created the pdfT<sub>E</sub>X programme in the first place, which introduced the micro-typographic extensions and made them available to the T<sub>E</sub>X world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#), [Thành 2008](#)

and elsewhere. I also thank him and the rest of the pdfTeX team, and more recently also the LuaTeX and XeTeX teams, for refuting the idea that TeX is dead, and for fixing the bugs I find.

*Harald Harders* has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. *Andreas Böhmann* has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. *Ulrich Dirr* has made numerous suggestion, especially concerning the new extensions of interword spacing adjustment and additional character kerning. *Georg Duffner* has patiently tested microtype under XeTeX and LuaTeX with his beautiful OpenType font EB Garamond<sup>13</sup>. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding, as well as to *Karl Karlsson* for providing settings for the Cyrillic T2A encoding, and to *Hendrik Vogt*, who made substantial improvements to the Computer Modern Roman italic settings. I thank *Loren B. Davis* for providing protrusion settings for OpenType versions of Palatino Linotype. I am also very much indebted to *Élie Roux*, who not only contributed the lua module in the first place, but also, together with *Philipp Gesang*, took care of updating it for the developments in LuaTeX land.

I thank *Philipp Lehman* for adding to his csquotes package the possibility to restore the original meanings of all activated characters, thus allowing for these characters to be used in the configuration files. *Peter Wilson* kindly provided a hook in his ledmac/ledpar packages, so that critical editions can finally also benefit from character protrusion. Likewise, *Donald Arseneau* patched his shapepar package to accommodate protrusion.

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order, quotes indicate TeX.SX user names): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulte*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dyballa*, *Morten Høgholm*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis*, *Sven Naumann*, *Ross Hetherington*, *Wiebke Petersen*, *Geoff Vallis*, *Steven E. Harris*, *Karl Berry*, *Peter Meier*, *Nathan Rosenblum*, *Wolfram Schaalo*, *Vasile Gaburici*, *Sveinung Heggen*, *Axel Berger*, *Colin Rourke*, *Maverick Woo*, *Silas S. Brown*, *Lars Rönnbäck*, *Christian Stark*, *Leo*, *Marcin Borkowski*, *hscm*, *George Gratzner*, *Josep Maria Font*, *Juan Acevedo*, *Heiko Oberdiek*, *Till A. Heilmann*, *Rolf Dieterich*, *Seamus Bradley*, *Meho R*, *Steffen Hoffmann*, *Scott Pakin*, *Maïeul Rouquette*, *Jonas Hogstrom*, *Gabriel Kerneis*, ‘RazorXsr’, *Sebastian Schubert*, ‘Dave’, *Giuseppe Palma*, *Stephan Stiller*, *Christopher Schramm*, ‘uli’, *Sam Mason*, ‘kleenstar’, ‘Henning’, *Ronnie Marks*, *David Carlisle*, ‘Max’, ‘HcN’, *Will Robertson*, ‘user11126’, *Ulrike Fischer*, ‘Daniel’, ‘ltcomdata’, *Reinhard Kotucha*, ‘jcr’, *Nils Anders Danielsson*, *Paolo Ney*, *Frank Mittelbach*, *Franz Wexler*, *Moritz Wemheuer*, ‘Andy N’, *Phelype Oleinik*, *Falk Hanisch*, *Markus Kohm*, *Paolo Polesana*, *Oliver Kopp*, *Hironori Kitagawa*, *Daniel Benjamin Miller*, *Md Ayquassar*, *Marcel Krüger*, *Ekkehart Schlicht*, ‘Canageek’, ‘dsedivec’ and ‘DORpapist’.

13 Available from CTAN at [pkg/ebgaramond](http://pkg/ebgaramond), including configuration files for microtype.

## 12 References

Hàn Thế Thành, ‘Micro-typographic extensions to the  $\text{\TeX}$  typesetting system’, Diss. Masaryk University Brno 2000, in: *TUGBoat*, vol. 21 (2000), no. 4, pp. 317–434. (Online at <http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf>)

Hàn Thế Thành, ‘Micro-typographic extensions of pdf $\text{\TeX}$  in practice’, in: *TUGBoat*, vol. 25 (2004), no. 1: ‘Proceedings of the Practical  $\text{\TeX}$  2004 Conference’, pp. 35–38. (Online at <http://www.tug.org/TUGboat/Articles/tb25-1/thanh.pdf>)

Hàn Thế Thành, ‘Font-specific issues in pdf $\text{\TeX}$ ’, in: *TUGBoat*, vol. 29 (2008), no. 1: ‘EuroBach $\text{\TeX}$  2007 Proceedings’, pp. 36–41. (Online at <http://www.tug.org/TUGboat/Articles/tb29-1/tb91thanh-fonts.pdf>)

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, Karl Berry, *The pdf $\text{\TeX}$  user manual*, 2 March 2020. (Available from CTAN at [/systems/doc/pdftex/manual/pdftex-a.pdf](#))

Karl Berry, *Fontname: Filenames for  $\text{\TeX}$  fonts*, July 2009. (Available from CTAN at [/info/fontname/fontname.pdf](#))

$\text{\LaTeX}$ 3 Project Team,  *$\text{\LaTeX}$ 2 $\epsilon$  font selection*, November 2020. (Available from CTAN at [/macros/latex/base/fntguide.pdf](#))

Will Robertson, *The fontspec package: Font selection for  $\text{\XeLaTeX}$  and  $\text{\LuaLaTeX}$* , 21 February 2020. (Available from CTAN at [pkg/fontspec](#))

Élie Roux, Khaled Hosny, Philipp Gesang, Ulrike Fischer, Marcel Krüger, *The luaotfload package*, 8 January 2021. (Available from CTAN at [pkg/luaotfload](#))

Carsten Schurig, Tobias Schlemmer, *The pdfcp $\text{\TeX}$ .sty package*, 10 June 2005. (Available from CTAN at [pkg/pdfcp \$\text{\TeX}\$](#) )

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at [pkg/soul](#)). See also Heiko Oberdiek’s extension of this package, `soulut $\text{\TeX}$ 8`, which adds Unicode support. (Available from CTAN at [pkg/soulut \$\text{\TeX}\$ 8](#))

## 13 Short history

The comprehensive list of changes can be found in appendix C. The following is a list of all changes relevant in the user land; bug and compatibility fixes are swept under the rug. Numbers in brackets indicate the relevant section in this manual.

### 2.8 (2020/12/07)

- New default font sets for expansion and spacing: ‘all $\text{\TeX}$ -nott’ [4, table 2]

### 2.7 (2017/07/07)

- Allow automatic expansion and letterspacing with  $\text{\LuaTeX}$  in DVI mode (aka. `dvilualatex`) [3.1, 3.3, table 1]
- Compatibility with  $\text{\LaTeX}$  2017/01/01 (fix warnings)

### 2.6 (2016/05/01)

- Support for  $\text{\LuaTeX} \geq 0.85$
- Improvements for tracking/letterspacing with  $\text{\LuaTeX}$  (Renderer=Basic no longer required)

- New font sets: ‘alltext-nott’, ‘allmath-nott’ [4, table 2]

## 2.5 (2013/03/13)

- Support for the `fontspec` package, viz. for OpenType fonts with LuaTeX and XeTeX
- Support for protrusion with XeTeX  $\geq 0.9997$
- Support for tracking/letterspacing with LuaTeX  $\geq 0.62$
- Allow context-sensitive setup with LuaTeX
- Info if protrusion settings are generic
- Protrusion settings for Latin Modern Roman (OpenType)
- Protrusion settings for Charis SIL (OpenType)
- Protrusion settings for Palatino Linotype (OpenType)

## 2.4 (2010/01/10)

- Protrusion settings for T2A encoded Minion

## 2.3e (2009/11/09)

- Support for the Cyrillic T2A encoding (protrusion, expansion, spacing)

## 2.3d (2009/03/27)

- New default for expansion option ‘step’: 1, if pdfTeX  $\geq 1.40$  [3.3]

## 2.3c (2008/11/11)

- Support for LuaTeX enabled by default

## 2.3 (2007/12/23)

- New key ‘outer kerning’ for `\SetTracking` to customise outer kerning [5.3]
- Adjust protrusion settings for tracking even if protrusion is not enabled
- New option ‘verbose=silent’ to turn all warnings into mere messages [3.5]
- The `letterspace` package also works with `eplain` or `miniltx` [7]

## 2.2 (2007/07/14)

- Improvements to tracking/letterspacing: retain kerning (pdfTeX  $\geq 1.40.4$ ); automatically adjust protrusion settings
- New key ‘no ligatures’ for `\SetTracking` to disable selected or all ligatures (pdfTeX  $\geq 1.40.4$ ) [5.3]
- New keys ‘spacing’ and ‘outer spacing’ for `\SetTracking` to customise interword spacing [5.3]
- Possibility to expand a font with different parameters (pdfTeX  $\geq 1.40.4$ ) [5.2]
- New optional argument for `\DisableLigatures` to disable selected ligatures [8]
- New command `\DeclareMicrotypeVariants` to specify variant suffixes [5.7]
- New command `\textmicrotypecontext` as a wrapper for `\microtypecontext` [6]
- Protrusion settings for Bitstream Letter Gothic

## 2.1 (2007/01/21)

- New command `\slig` to protect ligatures in letterspaced text [7]

## 2.0 (2007/01/14)

- Support for the new extensions of pdfTeX  $\geq 1.40$ : tracking/letterspacing, additional kerning, and adjustment of interword spacing (glue) (new commands `\SetTracking`, `\SetExtraKerning`, `\SetExtraSpacing`; new options ‘tracking’, ‘kerning’, ‘spacing’) [5.3, 5.4, 5.5]



- New commands `\textls` and `\lststyle` for letterspacing, new option ‘letterspace’ [3.4, 7]
- New option ‘babel’ for automatic micro-typographic adjustment to the selected language [3.5, 6]
- New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ [4, table 2]
- New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing [7]

**1.9e (2006/07/28)**

- New key ‘inputenc’ to specify the lists’ input encodings [5]
- Protrusion settings for Euler math fonts

**1.9d (2006/05/05)**

- Support for the Central European QX encoding (protrusion, inheritance)
- Protrusion settings for various Euro symbol fonts (Adobe, ITC, marvosym)
- Support for Unicode input in the configuration (inputenc/utf8)

**1.9c (2006/02/02)**

- Protrusion settings for URW Garamond

**1.9a (2005/12/05)**

- Defer setup until the end of the preamble
- Inside the preamble, `\microtypesetup` accepts all package options [3.6]
- Protrusion settings for T5 encoded Charter

**1.9 (2005/10/28)**

- New command `\DisableLigatures` to disable ligatures (pdfTeX  $\geq 1.30$ ) [8]
- New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands [6]
- New key ‘font’ to add single fonts to the font sets [4]
- New key ‘preset’ to set all characters to the specified value before loading the lists
- Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
- Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
- Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)

**1.8 (2005/06/23)**

- New command `\DeclareMicrotypeSetDefault` to declare the default font sets [4]
- New option ‘config’ to load a different configuration file [3.5]
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width [5.1]
- Renamed commands from `\..MicroType..` to `\..Microtype..`
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The ‘allmath’ font set also includes U encoding
- Support for protrusion with the `ledmac` package (pdfTeX  $\geq 1.30$ )

**1.7 (2005/03/23)**

- Possibility to specify ranges of font sizes in the set declarations [4, 5]
- New command `\LoadMicrotypeFile` to load a configuration file manually [5.7]
- New command `\Microtype@Hook` for font package authors [14.4.4]
- New option ‘verbose=errors’ to turn all warnings into errors



- Warning when running in draft mode

#### 1.6 (2005/01/24)

- New option ‘factor’ to influence protrusion resp. expansion of all characters of a font or font set [3.2, 5]
- When pdfTeX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled [3.1]
- Use e-TeX extensions, if available

#### 1.5 (2004/12/15)

- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled [3.1]
- New option ‘selected’ to enable selected expansion, default: false [3.3, 5.2]
- New default for expansion option ‘step’: 4 (min(stretch,shrink)/5) [3.3]
- Protrusion settings for Bitstream Charter

#### 1.4 (2004/11/12)

- Set up fonts independently from L<sup>A</sup>T<sub>E</sub>X font loading
- New option: ‘final’ [3.5]

#### 1.2 (2004/10/03)

- New font sets: ‘allmath’ and ‘basicmath’ [4, table 2]
- Protrusion settings for Computer Modern Roman math symbols
- Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond

#### 1.1 (2004/09/21)

- Protrusion settings for Adobe Minion
- New command: \DeclareCharacterInheritance [5.6]
- Characters may also be specified as octal or hexadecimal numbers [5]

#### 1.0 (2004/09/11)

- First CTAN release

## 14 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).  
 pdftex-def: Definitions specific to pdfTeX (microtype-pdftex.def).  
 xetex-def: Definitions specific to XeTeX (microtype-xetex.def).  
 luatex-def: Definitions specific to LuaTeX (microtype-luatex.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.

<sup>1</sup> `<package|letterspace>`

## 14.1 Preliminaries

`\MT@MT` This is us.

```
2 \def\MT@MT
3 <package> {microtype}
4 <letterspace> {letterspace}
```

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

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

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

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

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

```

54 \newcommand*\lslig[1]{#1}
55 <package>
56 }

```

These commands also have a starred version.

```

57 \def\DeclareMicrotypeSet#1#\@gobbletwo{
58 \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.

```

59 \@onlypreamble\DeclareMicrotypeSet
60 \@onlypreamble\UseMicrotypeSet
61 \@onlypreamble\DeclareMicrotypeSetDefault
62 \@onlypreamble\DisableLigatures
63 \@onlypreamble\DeclareMicrotypeVariants
64 \@onlypreamble\DeclareMicrotypeBabelHook

```

Don't load letterspace.

```

65 \expandafter\let\csname ver@letterspace.sty\endcsname\@empty

```

`\MT@old@cmd` The old command names had one more hunch.

```

66 \def\MT@old@cmd#1#2{%
67 \newcommand*#1{\MT@warning{%
68 \string#1 is deprecated. Please use\MessageBreak
69 \string#2 instead}%
70 \let #1#2#2}}
71 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
72 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
73 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
74 \MT@old@cmd\LoadMicroTypeFile \LoadMicrotypeFile
75 <package>

```

`\MT@warning` Communicate.

```

\MT@warning@nl 76 \def\MT@warning{\PackageWarning\MT@MT{
\MT@info 77 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
78 <package>
\MT@info@nl 79 \def\MT@info{\PackageInfo\MT@MT{
\MT@vinfo 80 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
\MT@error 81 \let\MT@vinfo\@gobble
\MT@warn@err 82 \def\MT@error{\PackageError\MT@MT{
83 \def\MT@warn@err#1{\MT@error{#1}{%
84 This error message appears because you loaded the \MT@MT'\MessageBreak
85 package with the option `verbose=errors'. Consult the documentation\MessageBreak
86 in \MT@MT.pdf to find out what went wrong.}}

```

### 14.1.1 Debugging

`\tracingmicrotype` Cases for `\tracingmicrotype`:

```

\MT@dinfo 0: almost none
\MT@dinfo@nl 1: + sets & lists
2: + heirs
3: + slots
4: + factors

```

```

87 <debug>
88 \MT@warning@nl{This is the debug version}
89 \newcount\tracingmicrotype

```

\tracingmicrotypeinpdf

Cases for \tracingmicrotypeinpdf:

- 1: show new fonts
- 2: + show known fonts

*[If microtype.sty had been generated with the ‘debug’ option,  
this method would be demonstrated here.]*

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

```

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

```

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

\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).

```

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

```

### 14.1.2 Requirements

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

- 0: miniltx
- 1: eplain
- 2: L<sup>A</sup>T<sub>E</sub>X

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

```

125 <plain>
126 \def\MT@plain{2}
127 \ifx\documentclass\undefined
128   \def\MT@plain{1}
129   \def\hmodebgroup{\leavevmode\bgroup}
130   \def\nfss@text#1{{\mbox{#1}}}
131   \let\@typeset@protect\relax
132   \ifx\eplain\undefined
133     \def\MT@plain{0}
134     \def\PackageWarning#1#2{%
135       \begingroup
136         \newlinechar=10 %
137         \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
138         \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
139       \endgroup
140     }
141     \def\on@line{ on input line \the\inputlineno}
142     \def\@spaces{\space\space\space\space}
143   \fi
144 \fi

```

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

```

145 \def\MT@requires@latex#1{%
146   \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
147 }
148 <plain>

```

For definitions that depend on e-TeX features.

```

149 \ifcase 0%
150   \ifx\TeXversion\undefined 1\else
151     \ifx\TeXversion\relax 1\else
152       \ifcase\TeXversion 1\fi
153     \fi
154   \fi
155 \else
156   \catcode\^^Q=9 \catcode\^^X=14
157 \fi
158 <letterspace>^^Q\MT@warning@nl{This package requires the etex extensions.
159 <letterspace>^^Q \MessageBreak Exiting}\MT@restore@catcodes\endinput
160 <debug>\MT@info@nl{0}{this is
161 <debug>^^Q not
162 <debug> etex}

```

We check whether we are running pdf<sub>T</sub>E<sub>X</sub>, X<sub>Y</sub><sub>T</sub>E<sub>X</sub>, or Lua<sub>T</sub>E<sub>X</sub>, and load the appropriate definition file.

`\MT@clear@options` If we are using neither of these engines, we disable everything and exit.

```

163 \def\MT@clear@options{%
164 <plain> \MT@requires@latex1{%
165   \AtEndOfPackage{\let\@unprocessedoptions\relax\MT@restore@catcodes}%
166   \let\CurrentOption\@empty
167 <package> \let\MT@endinput\endinput
168 <plain> }\relax
169 }

```

A hack circumventing the T<sub>E</sub>X Live 2004 hack which undefines the pdfT<sub>E</sub>X primitives in the format in order to hide the fact that pdfT<sub>E</sub>X is being run from the user. This has been *fixed* in T<sub>E</sub>X Live 2005.

```
170 \ifx\normalpdfptextversion\@undefined \else
171   \let\pdfptextversion \normalpdfptextversion
172   \let\pdfptextrevision\normalpdfptextrevision
173   \let\pdfptextoutput   \normalpdfptextoutput
174 \fi
```

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

```
\MT@engine@toold 175 \let\MT@engine\relax
176 <letterspace>\def\MT@engine@toold{0}
177 \ifx\pdfptextversion\@undefined \else
178   \ifx\pdfptextversion\relax \else
179     \def\MT@engine{pdf}
180 <letterspace>   \let\MT@pdf@or@lua\@firstoftwo
181 <letterspace>   \ifnum\pdfptextversion > 139 \def\MT@engine@toold{1}\fi
182 \fi
183 \fi
184 \ifx\directlua\@undefined \else
185   \ifx\directlua\relax \else
186     \def\MT@engine{lua}
```

Since approx. LuaT<sub>E</sub>X 0.80, \pdfptextversion is let to \luatextversion, so that we would be fooled to think that pdfT<sub>E</sub>X is too old.

```
187 <*letterspace>
188   \let\MT@pdf@or@lua\@secondoftwo
189   \ifnum\luatextversion < 62 \def\MT@engine@toold{0}
190   \else
191     \def\MT@engine@toold{1}
192     \let\MT@lua\directlua
193     \ifnum\luatextversion > 84
194       \let\pdfptextoutput\outputmode
195       \let\pdfptextprotrudechars\protrudechars
196       \let\pdfptextadjustspacing\adjustspacing
197     \fi
198   \fi
199 </letterspace>
200 \fi
201 \fi
202 <*package>
203 \ifx\MT@engine\relax
204   \ifx\XeTeXversion\@undefined \else
205     \ifx\XeTeXversion\relax \else
206       \def\MT@engine{xe}
207     \fi
208   \fi
209 \fi
210 </package>
211 </package|letterspace>
```

\MT@pdfptext@no pdfT<sub>E</sub>X'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 pdfT<sub>E</sub>X we're using, if any. \MT@pdfptext@no will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for pdfT<sub>E</sub>X:

- 0: not running pdfT<sub>E</sub>X
- 1: pdfT<sub>E</sub>X (< 0.14f)
- 2: + micro-typographic extensions (0.14f,g)
- 3: + protrusion relative to 1em (≥ 0.14h)

- 4: + automatic font expansion; protrusion no longer has to be set up first; scale factor fixed to 1000; default `\efcode = 1000` ( $\geq 1.20$ )
- 5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` ( $\geq 1.30$ )
- 6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`<sup>14</sup>; `\pdftracingfonts`; always e-TeX ( $\geq 1.40$ )
- 7: + `\letterspacefont` doesn't disable ligatures and kerns; `\pdfcopyfont` ( $\geq 1.40.4$ )

```

212 (*pdfTeX-def)
213 (debug)\MT@info@nl{0}{this is pdfTeX \the\pdfTeXversion(\pdfTeXrevision)}
214 \def\MT@pdfTeX@no{7}
215 \ifnum\pdfTeXversion = 140
216   \ifnum\pdfTeXrevision < 4
217     \def\MT@pdfTeX@no{6}
218   \fi
219 \else
220   \ifnum\pdfTeXversion < 140
221     \def\MT@pdfTeX@no{5}
222     \ifnum\pdfTeXversion < 130
223       \def\MT@pdfTeX@no{4}
224       \ifnum\pdfTeXversion < 120
225         \def\MT@pdfTeX@no{3}
226         \ifnum\pdfTeXversion = 14
227           \ifnum \expandafter`\pdfTeXrevision < `h
228             \def\MT@pdfTeX@no{2}
229           \ifnum \expandafter`\pdfTeXrevision < `f
230             \def\MT@pdfTeX@no{1}
231           \fi
232         \fi
233       \else
234         \ifnum\pdfTeXversion < 14
235           \def\MT@pdfTeX@no{1}
236         \fi
237       \fi
238     \fi
239   \fi
240 \fi
241 \fi
242 (debug)\MT@info@nl{0}{pdfTeX no.: \MT@pdfTeX@no}
243 (pdfTeX-def)

```

`\MT@xetex@no`     XeTeX supports character protrusion since version 0.9997.

```

244 (*xetex-def)
245 (debug)\MT@info@nl{0}{this is xetex (\the\XeTeXversion\XeTeXrevision)}
246 \ifdim 0\XeTeXrevision pt < 0.9997pt
247   \def\MT@xetex@no{1}
248 \else
249   \def\MT@xetex@no{2}
250 \fi
251 (debug)\MT@info@nl{0}{xetex no.: \MT@xetex@no}
252 (xetex-def)

```

`\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 ( $\geq 0.36$ )

---

<sup>14</sup> This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.



- 3: + `\letterspacefont`; non-automatic expansion doesn't work anymore, and automatic expansion in DVI mode is realised by modifying the tracking, not the glyphs<sup>15</sup> ( $\geq 0.62$ )
- 4: + almost all of the pdfTeX primitives have been renamed ( $\geq 0.85$ )
- 5: + default `\efcode = 1000`; `\protrusionboundary` [not yet supported] ( $\geq 0.90$ )
- 6: + `\glet` ( $\geq 1.10$ )

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).

```

253 <lua-def>
254 <debug> \MT@info@nl{this is luatex (\the\luatexversion)}

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

255 \let\MT@lua\directlua
256 \def\MT@luatex@no{6}
257 \ifnum\luatexversion<110
258   \def\MT@luatex@no{5}
259   \ifnum\luatexversion<90
260     \def\MT@luatex@no{4}
261     \ifnum\luatexversion<85
262       \def\MT@luatex@no{3}
263       \ifnum\luatexversion<62
264         \def\MT@luatex@no{2}
265         \ifnum\luatexversion<36
266           \def\MT@lua{\directlua0}
267           \def\MT@luatex@no{1}
268         \fi
269       \fi
270     \fi
271   \fi
272 \fi

273 <debug> \MT@info@nl{0}{luatex no.: \MT@luatex@no}
274 </lua-def>

275 <pdf-def|xetex-def|letterspace>
276 \ifnum
277 <pdf-def|xetex-def> \csname MT@\MT@engine tex@no\endcsname < 2
278 <letterspace> \MT@engine@toold=\z@
279 \MT@warning@nl{You
280 <letterspace>
281   \ifx\MT@engine\relax
282     don't seem to be using pdfTeX or LuaTeX.\MessageBreak
283     Try running 'pdfTeX' or 'LuaTeX' instead of.\MessageBreak
284     \ifx\XeTeXversion\undefined\else xe\fi tex'%
285   \else
286 </letterspace>
287     are using a \MT@engine tex version older than
288 <pdf-def>         0.14f%
289 <xetex-def>       0.9997%
290 <letterspace>     \MT@pdf@or@lua{1.40}{0.62}%
291     .\MessageBreak
292     '\MT@MT' does not work with this version.\MessageBreak
293     Please install a newer version of \MT@engine tex%
294 <letterspace>     \fi
295     .\MessageBreak I will quit now}
296 \MT@clear@options
297 \endinput\fi
298 </pdf-def|xetex-def|letterspace>

```

15 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).

Still there? Then we can begin: We need the `keyval` package, including the ‘new’ `\KV@sp@def` implementation.

```
299 <package>|letterspace>
300 \RequirePackage{keyval}[1997/11/10]
301 <package>
```

`\MT@toks` We need a token register.

```
302 \newtoks\MT@toks
```

`\ifMT@if@` A scratch if.

```
303 \newif\ifMT@if@
```

### 14.1.3 Declarations

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

```
\ifMT@expansion 304 \newif\ifMT@protrusion
\ifMT@auto        305 \newif\ifMT@expansion
\ifMT@selected    306 \newif\ifMT@auto
\ifMT@noligatures 307 \newif\ifMT@selected
\ifMT@draft       308 \newif\ifMT@noligatures
\ifMT@spacing     309 \newif\ifMT@draft
\ifMT@kerning     310 \newif\ifMT@spacing
\ifMT@tracking    311 \newif\ifMT@kerning
\ifMT@babel       312 \newif\ifMT@tracking
\ifMT@babel       313 \newif\ifMT@babel
```

`\ifMT@babel` [This line intentionally left blank.]

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

```
\MT@ex@level 314 \let\MT@pr@level\tw@
\MT@pr@factor 315 \let\MT@ex@level\tw@
\MT@ex@factor 316 \let\MT@pr@factor\@m
\MT@sp@factor 317 \let\MT@ex@factor\@m
\MT@kn@factor 318 \let\MT@sp@factor\@m
\MT@kn@factor 319 \let\MT@kn@factor\@m
```

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

```
\MT@sp@unit
\MT@kn@unit 320 \let\MT@pr@unit\@empty
321 \let\MT@sp@unit\m@ne
322 \def\MT@kn@unit{1em}
```

`\MT@stretch` Expansion settings.

```
\MT@shrink 323 \let\MT@stretch\m@ne
\MT@step    324 \let\MT@shrink \m@ne
325 \let\MT@step \m@ne
```

`\MT@pr@min` Minimum and maximum values allowed by pdfTeX.

```
\MT@pr@max 326 \def\MT@pr@min{-\@m}
\MT@ex@min 327 \let\MT@pr@max\@m
\MT@ex@max 328 \let\MT@ex@min\z@
\MT@sp@min 329 \let\MT@ex@max\@m
\MT@sp@max 330 \def\MT@sp@min{-\@m}
\MT@kn@min 331 \let\MT@sp@max\@m
\MT@kn@max 332 \def\MT@kn@min{-\@m}
\MT@tr@min 333 \let\MT@kn@max\@m
\MT@tr@max 334 <package>
335 \def\MT@tr@min{-\@m}
336 \let\MT@tr@max\@m
337 <package>
```

`\MT@factor@default` Default factor.

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

```

\MT@stretch@default      Default values for expansion.
\MT@shrink@default      339 \def\MT@stretch@default{20 }
                        340 \def\MT@shrink@default{20 }

\MT@letterspace          Default value for letterspacing (in thousandths of 1 em).
\MT@letterspace@default  341 </package>
                        342 \let\MT@letterspace\m@ne
                        343 \def\MT@letterspace@default{100}
                        344 <*package>

\ifMT@document          Our private test whether we're still in the preamble.
                        345 \newif\ifMT@document
                        346 </package>
                        347 </package|letterspace>

```

#### 14.1.4 Auxiliary macros

```

\MT@requires@pdftex      For definitions that depend on a particular pdfTeX resp. LuaTeX version.
\MT@requires@luatex      348 <*pdftex-def| luatex-def>
                        349 \def
                        350 <pdftex-def> \MT@requires@pdftex%
                        351 <luatex-def> \MT@requires@luatex%
                        352 #1{\ifnum
                        353 <pdftex-def> \MT@pdftex@no
                        354 <luatex-def> \MT@luatex@no
                        355 <#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi>
                        356 <luatex-def&debug>\MT@requires@luatex4{\MT@lua{tex.enableprimitives('pdf',{'tracingfonts'})}}\relax
                        357 <pdftex-def&debug>\MT@requires@pdftex6{
                        358 <debug>\pdftracingfonts=1
                        359 <pdftex-def&debug>}\relax
                        360 </pdftex-def| luatex-def>

```

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<sup>A</sup>T<sub>E</sub>X, we load the luatexbase package.

```

361 <*luatex-def>
362 \ifl@t@r\fmtversion{2016/01/01}\relax{\RequirePackage{luatexbase}}

```

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

```

363 \ifl@t@r\fmtversion{2017/01/01}\relax{\RequirePackage{luaotfload}}
364 \MT@lua{require("microtype")}
365 </luatex-def>

```

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

```

366 <*luafile>
367
368 function microtype.info(...)
369   luatexbase.module_info("microtype",...)
370 end
371
372 local find      = string.find
373 local match     = string.match
374 local tex_write = tex.write
375
376 local catpackage
377 if luatexbase.registernumber then
378   catpackage = luatexbase.registernumber("catcodetable@atletter") -- LaTeX
379 else

```

```

380 catpackage = luatexbase.catcodetables.CatcodeTableLaTeXAtLetter -- luatexbase
381 end
382 function microtype.sprint (...)
383   tex.sprint(catpackage, ...)
384 end
385
386 /luafile

```

To be continued, but first back to primitives.

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

```

387 <luatex-def> \MT@requires@luatex6{\let\MT@glet\glet}\relax
388 <*package|letterspace>
389 \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.

```

\MT@exp@gcs
390 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
391 <*package>
392 \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 393 \def\MT@def@n{\MT@exp@cs\def}
394 \def\MT@gdef@n{\MT@exp@gcs\gdef}

```

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

```

\MT@xdef@n 395 </package>
396 \def\MT@edef@n{\MT@exp@cs\edef}
397 <*package>
398 \def\MT@xdef@n{\MT@exp@gcs\xdef}

```

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

```

\MT@glet@nc 399 \def\MT@let@nc{\MT@exp@cs\let}
400 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

```

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

```

401 </package>
402 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
403 <*package>

```

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

```

\MT@glet@nn 404 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
405 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

```

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

```

406 \def\MT@@font{\expandafter\string\MT@font}

```

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

```

407 </package>
408 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}

```

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

```

409 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
410 <*package>

```

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

```

411 \def\MT@exp@two@n#1#2#3{%
412   \expandafter\expandafter\expandafter
413   #1\expandafter\expandafter\expandafter
414   {\expandafter#2\expandafter}\expandafter{#3}}

```

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

<code>\MT@ifdefined@c@T</code>	Wrapper for testing whether command resp. <code>\csname</code> sequence is defined. If we
<code>\MT@ifdefined@c@TF</code>	are running e-TeX, we will use its primitives <code>\ifdefined</code> and <code>\ifcsname</code> , which
<code>\MT@ifdefined@n@T</code>	decreases memory use substantially.
<code>\MT@ifdefined@n@TF</code>	<pre> 415 \def\MT@ifdefined@c@T#1{% 416 ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi 417 ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi 418 } 419 &lt;/package&gt; 420 \def\MT@ifdefined@c@TF#1{% 421 ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi 422 &lt;package&gt;^^Q \ifx#1\@undefined 423 &lt;package&gt;^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi 424 } 425 \def\MT@ifdefined@n@T#1{% 426 ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi 427 &lt;package&gt;^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax 428 &lt;package&gt;^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi 429 } 430 \def\MT@ifdefined@n@TF#1{% 431 ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi 432 &lt;package&gt;^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax 433 &lt;package&gt;^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi 434 } 435 &lt;*package&gt; </pre>
<code>\MT@detokenize@n</code>	Translate a macro into a token list. With e-TeX, we can use <code>\detokenize</code> . We also
<code>\MT@detokenize@c</code>	need to remove the last trailing space; and only the last one – therefore the fiddling
<code>\MT@rem@last@space</code>	(and the <code>\string</code> isn't perfect, of course).
	<pre> 436 \def\MT@detokenize@n#1{% 437 ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil 438 ^^Q \string#1% 439 } 440 \def\MT@detokenize@c#1{% 441 ^^X \MT@exp@one@n\MT@detokenize@n#1% 442 ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil 443 } 444 \def\MT@rem@last@space#1 #2{#1% 445 \ifx\@nil#2\else \space 446 \expandafter\MT@rem@last@space\expandafter#2\fi 447 } </pre>
<code>\MT@ifempty</code>	Test whether argument is empty.
	<pre> 448 &lt;/package&gt; 449 \begingroup 450 \catcode`\%=12 451 \catcode`\&amp;=14 452 \gdef\MT@ifempty#1{% 453 \if %#1%&amp; 454 \expandafter\@firstoftwo 455 \else 456 \expandafter\@secondoftwo 457 \fi 458 } 459 \endgroup 460 &lt;*package&gt; </pre>
<code>\MT@ifint</code>	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 <code>letterspace</code> option).
	<pre> 461 &lt;/package&gt; 462 &lt;/package letterspace&gt; 463 &lt;pdfTeX-def&gt;\MT@requires@pdftex6{ 464 &lt;letterspace&gt;\MT@pdf@or@lua{ </pre>

```

465 <*pdftex-def|letterspace>
466 \def\MT@ifint#1{%
467   \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
468   \expandafter\@secondoftwo
469   \else
470   \expandafter\@firstoftwo
471   \fi
472 }
473 }{
474 </pdftex-def|letterspace>
475 <*pdftex-def|xetex-def|letterspace>
476 \def\MT@ifint#1{%
477   \if!\ifnum9<1#1!\else?\fi
478   \expandafter\@firstoftwo
479   \else
480   \expandafter\@secondoftwo
481   \fi
482 }
483 </pdftex-def|xetex-def|letterspace>
484 <pdftex-def|letterspace>}
485 <luatex-def>\def\MT@ifint#1{\csname\MT@lua{microtype.if_int}([#1])\endcsname}
486 <*luafile>
487 local function if_int(s)
488   if find(s,"^-[0-9]+ *$") then
489     tex_write("@firstoftwo")
490   else
491     tex_write("@secondoftwo")
492   end
493 end
494 microtype.if_int = if_int
495
496 </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.)

```

497 <*pdftex-def>
498 \MT@requires@pdftex6{
499 \def\MT@ifdimen#1{%
500   \ifcase\pdfmatch{^[0-9]+([.][0-9]+)?|.[0-9]+}%
501   (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
502   \expandafter\@secondoftwo
503   \else
504   \expandafter\@firstoftwo
505   \fi
506 }
507 }{
508 </pdftex-def>
509 <*pdftex-def|xetex-def>
510 \def\MT@ifdimen#1{%
511   \setbox\z@=\hbox{%
512     \MT@count=1#1\relax
513     \ifnum\MT@count=\@ne
514       \aftergroup\@secondoftwo
515     \else
516       \aftergroup\@firstoftwo
517     \fi
518   }%
519 }
520 </pdftex-def|xetex-def>
521 <pdftex-def>}
522 <luatex-def>\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen}([#1])\endcsname}
523 <*luafile>
524 local function if_dimen(s)
525   if (find(s,"^-[0-9]+(%a*) *$") or
526       find(s,"^-[0-9]*[.][0-9]+(%a*) *$")) then

```

```

527     tex_write("@firstoftwo")
528   else
529     tex_write("@secondoftwo")
530   end
531 end
532 microtype.if_dimen = if_dimen
533
534 </luafile>

\MT@ifdim    Compare floating point numbers.

535 <*package>
536 \def\MT@ifdim#1#2#3{%
537   \ifdim #1\p@ #2 #3\p@
538     \expandafter\@firstoftwo
539   \else
540     \expandafter\@secondoftwo
541   \fi
542 }
543 </package>

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

544 <pdfTex-def|XeTeX-def>
545 <pdfTex-def>\MT@requires@pdfTex5{
546 \def\MT@ifstreq#1#2{%
547   <pdfTex-def> \ifnum\pdfstrcmp{#1}{#2}=\z@
548   <XeTeX-def> \ifnum\strcmp{#1}{#2}=\z@
549     \expandafter\@firstoftwo
550   \else
551     \expandafter\@secondoftwo
552   \fi
553 }
554 </pdfTex-def|XeTeX-def>
555 <*pdfTex-def>
556 }{
557 \def\MT@ifstreq#1#2{%
558   \edef\MT@res@a{#1}%
559   \edef\MT@res@b{#2}%
560   \ifx\MT@res@a\MT@res@b
561     \expandafter\@firstoftwo
562   \else
563     \expandafter\@secondoftwo
564   \fi
565 }
566 }
567 </pdfTex-def>
568 <LaTeX-def>\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq}([#1],[#2])\endcsname}
569 <*luafile>
570 local function if_str_eq(s1, s2)
571   if s1 == s2 then
572     tex_write("@firstoftwo")
573   else
574     tex_write("@secondoftwo")
575   end
576 end
577 microtype.if_str_eq = if_str_eq
578
579 </luafile>

\MT@xadd    Add item to a list.

580 <*package>
581 \def\MT@xadd#1#2{%
582   \ifx#1\relax
583     \xdef#1{#2}%
584   \else
585     \xdef#1{#1#2}%

```

```

586 \fi
587 }

\MT@xaddb      Add item to the beginning.
588 \def\MT@xaddb#1#2{%
589 \ifx#1\relax
590 \xdef#1{#2}%
591 \else
592 \xdef#1{#2#1}%
593 \fi
594 }
595 <package>

\MT@map@clist@n      Run <#2> on all elements of the comma list <#1>. This and the following is modelled
\MT@map@clist@c      after LATEX3 commands.
\MT@map@clist@      596 <package>|letterspace
\MT@clist@function  597 \def\MT@map@clist@n#1#2{%
\MT@clist@break      598 \ifx\@empty#1\else
                    599 \def\MT@clist@function##1{#2}%
                    600 \MT@map@clist@#1,\@nil,\@nnil
                    601 \fi
                    602 }

                    603 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
                    604 \def\MT@map@clist@c#1,%
                    605 \ifx\@nil#1%
                    606 \expandafter\MT@clist@break
                    607 \fi
                    608 \MT@clist@function{#1}%
                    609 \MT@map@clist@
                    610 }
                    611 \let\MT@clist@function\@gobble
                    612 \def\MT@clist@break#1\@nnil{}
                    613 <package>

\MT@map@tlist@n      Execute <#2> on all elements of the token list <#1>. \MT@tlist@break can be used
\MT@map@tlist@c      to jump out of the loop.
\MT@map@tlist@      614 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break      615 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
                    616 \def\MT@map@tlist@#1#2{%
                    617 \ifx\@nnil#2\else
                    618 #1{#2}%
                    619 \expandafter\MT@map@tlist@
                    620 \expandafter#1%
                    621 \fi
                    622 }
                    623 \def\MT@tlist@break#1\@nnil{\fi}

\ifMT@inlist@      Test whether item <#1> is in comma list <#2>. Using \pdfmatch would be slower.
\MT@in@clist      624 \newif\ifMT@inlist@
                    625 \def\MT@in@clist#1#2{%
                    626 \def\MT@res@a#1,#1,##2##3\@nnil{%
                    627 \ifx##2\@empty
                    628 \MT@inlist@false
                    629 \else
                    630 \MT@inlist@true
                    631 \fi
                    632 }%
                    633 \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
                    634 }

\MT@rem@from@clist      Remove item <#1> from comma list <#2>. This is basically \@removeelement from
                        ltcntrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
                    635 \def\MT@rem@from@clist#1#2{%

```



```

636 \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
637 \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
638 \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
639 }

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

640 \def\MT@in@tlist#1#2{%
641   \MT@inlist@false
642   \def\MT@res@a{#1}%
643   \MT@map@tlist@c#2\MT@in@tlist@
644 }
645 \def\MT@in@tlist@#1{%
646   \edef\MT@res@b{#1}%
647   \ifx\MT@res@a\MT@res@b
648     \MT@inlist@true
649     \expandafter\MT@tlist@break
650   \fi
651 }

\MT@in@rlist    Test whether size \MT@size is in a list of ranges. Store the name of the list in
\MT@in@rlist@    \MT@size@name
\MT@in@rlist@@ 652 \def\MT@in@rlist#1{%
\MT@size@name 653   \MT@inlist@false
654   \MT@map@tlist@c#1\MT@in@rlist@
655 }
656 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
657 \def\MT@in@rlist@@#1#2#3{%
658   \MT@ifdim{#2}=\m@ne{%
659     \MT@ifdim{#1}=\MT@size
660     \MT@inlist@true
661     \relax
662   }%
663   \MT@ifdim\MT@size<{#1}\relax{%
664     \MT@ifdim\MT@size<{#2}%
665     \MT@inlist@true
666     \relax
667   }%
668 }%
669 \ifMT@inlist@
670   \def\MT@size@name{#3}%
671   \expandafter\MT@tlist@break
672 \fi
673 }

\MT@loop    This is the same as LATEX's \loop, which we mustn't use, since this could confuse an
\MT@iterate  outer \loop in the document.
\MT@repeat  674 </package>
675 \def\MT@loop#1\MT@repeat{%
676   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
677   \MT@iterate \let\MT@iterate\relax
678 }
679 \let\MT@repeat\fi

\MT@while@num    Execute <#3> from <#1> up to (excluding) <#2> (much faster than LATEX's \@whilenum).
680 \def\MT@while@num#1#2#3{%
681   \@tempcnta#1\relax
682   \MT@loop #3%
683   \advance\@tempcnta \@ne
684   \ifnum\@tempcnta < #2\MT@repeat
685 }
686 </package>|letterspace

```

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

```

687 <letterspace>\MT@pdf@or@lua{\let\MT@if@luaotf@font\@secondoftwo}{
688 <luatex-def|letterspace>\def\MT@if@luaotf@font{\csname\MT@lua{%
689 <luatex-def> microtype.if_luaotf_font()
690 <*luafile|letterspace>
691 <luafile>local function if_luaotf_font()
692   local thefont = font.getfont(font.current())
693   if thefont and ( thefont.format == "opentype" or thefont.format == "truetype" )
694     then tex.write("@firstoftwo")
695     else tex.write("@secondoftwo")
696   end
697 <luafile>end
698 <luafile>microtype.if_luaotf_font = if_luaotf_font
699 <luafile>
700 </luafile|letterspace>
701 <luatex-def|letterspace> }\endcsname
702 <luatex-def|letterspace>}
703 <letterspace>}

```

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

```

704 <pdfTeX-def|letterspace>\def\MT@do@font{\MT@while@num\z@\cc@lvi}

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

705 <*luatex-def>
706 \def\MT@do@font#1{%
707   \MT@if@luaotf@font{%
708     \def\MT@dofont@function{#1}%
709     \MT@lua{microtype.do_font()}%
710   }{\MT@while@num\z@\cc@lvi{#1}}%
711 }
712 </luatex-def>

```

This is the `lua` function, which is much faster than looping through all glyphs in `TeX`. Legacy fonts (which this function should never work on) don't contain a `v.index` field.

```

713 <*luafile>
714 local function do_font()
715   local thefont = font.getfont(font.current())
716   if thefont then
717     for i,v in next,thefont.characters do
718       if v.index == nil or v.index > 0 then
719         microtype.sprint([[ \@tempcnta=]]..i..[[\relax\MT@dofont@function]])
720       end
721     end
722   end
723 end
724 microtype.do_font = do_font
725
726 </luafile>

```

The `XYTeX` variant (it's slow ...!).

```

727 <*xetex-def>
728 \def\MT@do@font#1{%
729   \@tempcnta=\z@
730   \MT@loop
731     \iffontchar\MT@font\@tempcnta #1\fi
732     \advance\@tempcnta\@ne
733     \ifnum\@tempcnta < \XeTeXlastfontchar\MT@font \MT@repeat
734   }
735 </xetex-def>
736 <*package>

```

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

slightly faster.

```
737 \newcount\MT@count
738 \def\MT@increment#1{%
739 ^^X \edef#1{\number\numexpr #1 + 1\relax}%
740 ^^Q \MT@count=#1\relax
741 ^^Q \advance\MT@count \@ne
742 ^^Q \edef#1{\number\MT@count}%
743 }
```

`\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.

```
744 \def\MT@scale#1#2#3{%
745 ^^Q \multiply #1 #2\relax
746 \ifnum #3 = \z@
747 ^^X #1=\numexpr #1 * #2\relax
748 \else
749 ^^X #1=\numexpr #1 * #2 / #3\relax
750 ^^Q \divide #1 #3\relax
751 \fi
752 }
```

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

```
753 \def\MT@abbr@pr{protrusion}
754 \def\MT@abbr@ex{expansion}
755 \def\MT@abbr@pr@c{protrusion codes}
756 \def\MT@abbr@ex@c{expansion codes}
757 \def\MT@abbr@pr@inh{protrusion inheritance}
758 \def\MT@abbr@ex@inh{expansion inheritance}
759 \def\MT@abbr@n1{noligatures}
760 \def\MT@abbr@sp{spacing}
761 \def\MT@abbr@sp@c{interword spacing codes}
762 \def\MT@abbr@sp@inh{interword spacing inheritance}
763 \def\MT@abbr@kn{kerning}
764 \def\MT@abbr@kn@c{kerning codes}
765 \def\MT@abbr@kn@inh{kerning inheritance}
766 \def\MT@abbr@tr{tracking}
767 \def\MT@abbr@tr@c{tracking amount}
```

`\MT@abbr@tr` These we also need the other way round.  
`\MT@rbba@protrusion`  
`\MT@rbba@expansion`

```
768 \def\MT@rbba@protrusion{pr}
769 \def\MT@rbba@expansion{ex}
770 \def\MT@rbba@spacing{sp}
771 \def\MT@rbba@kerning{kn}
772 \def\MT@rbba@tracking{tr}
```

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

```
773 \def\MT@features{pr,ex,sp,kn,tr}
774 \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 command 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.

```
775 \def\MT@is@feature#1#2{%
776 \MT@in@clist{#1}\MT@features@long
777 \ifMT@inlist@
778 \expandafter\@firstofone
779 \else
780 \MT@error{`#1' is not an available micro-typographic\MessageBreak
```

```

781     feature. Ignoring #2}{Available features are: `~\MT@features@long'.}%
782     \expandafter\@gobble
783     \fi
784 }

```

### 14.1.5 Compatibility

For the record, the following L<sup>A</sup>T<sub>E</sub>X kernel commands will be modified by microtype:

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent` (all in section 14.2.9)
- `\showhyphens` (in section 14.4.6)

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.

```

785 \ifl@aded{tex}{wordcount}{%
786   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
787     Disabling `~\MT@MT', since it wouldn't work}%
788   \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.

```

789 \ifclassloaded{minimal}{%
790   \MT@warning@nl{Detected the `minimal' class.\MessageBreak
791     Expect lots of warnings and some malfunctions.\MessageBreak
792     You might want to use a proper class instead}%
793 }\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.

```

794 </package>
795 <*package|letterspace>
796 <plain>\MT@requires@latex1{
797 \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.

```

798 \def\MT@addto@setup{\g@addto@macro\MT@setup@

```

Don't hesitate with `miniltx`.

```

799 <plain>}{\let\MT@addto@setup~\firstofone}

```

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

```

800 \def\MT@with@package@T#1{\ifpackageloaded{#1}\@firstofone~\@gobble}
801 </package|letterspace>
802 <*package>

```

`\MT@with@babel@and@T` L<sup>A</sup>T<sub>E</sub>X's `\ifpackagewith` ignores the class options.

```

803 \def\MT@with@babel@and@T#1{%
804   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
805     \expandtwoargs\MT@in@clist{#1}
806     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
807     \ifMT@in@list~\expandafter\@gobble~\fi
808   }\@gobble
809 }

```

`\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 XeTeX. The successor packages `eledmac` and `reledmac` are also supported.

```

810 </package>
811 <pdfTeX-def>\MT@requires@pdfTeX5{
812 <*pdfTeX-def|luaTeX-def|xetex-def>
813   \def\MT@ledmac@setup{%
814     \ifMT@protrusion
815       \MT@ifdefined@c@TF\l@dunhbox@line{%

```

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

```

816       \MT@info@nl{Patching ((r)e)ledmac to enable character protrusion}%
817       \let\MT@led@unhbox@line\l@dunhbox@line
818       \renewcommand*{\l@dunhbox@line}[1]{%
819         \ifhbox##1%
820           \kern\leftmarginkern##1%
821           \expandafter\MT@led@unhbox@line\expandafter##1\expandafter
822           \kern\rightmarginkern##1%
823         \fi
824       }%
825     }%
826     \MT@warning@nl{%
827       Character protrusion in paragraphs with line\MessageBreak
828       numbering will only work if you update ledmac,\MessageBreak
829       or use one of its successors, eledmac or reledmac}%
830   }%
831   \fi
832 }
833 </pdfTeX-def|luaTeX-def|xetex-def>
834 <*pdfTeX-def>
835 }{
836   \def\MT@ledmac@setup{%
837     \ifMT@protrusion
838       \MT@warning@nl{%
839         The pdfTeX version you are using does not allow\MessageBreak
840         character protrusion in paragraphs with line\MessageBreak
841         numbering by the `((r)e)ledmac' package.\MessageBreak
842         Upgrade pdfTeX to version 1.30 or later}%
843     \fi
844   }
845 }
846 </pdfTeX-def>

```

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 `\#`.

```

847 <*package|letterspace>
848 <*package>
849 \def\MT@restore@p@h{\chardef\%~\% \chardef\#~\# }

```

`\ifMT@xunicode`

Two new conditionals for use with XeTeX or LuaTeX.

`\ifMT@fontspec`

```

850 \newif\ifMT@xunicode
851 \MT@with@package@T{xunicode}\MT@xunicodetrue

```

```

852 \newif\ifMT@fontspec
853 \MT@with@package@T{fontspec}\MT@fontspectrue

```

`\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
854 \let\MT@maybe@gobble@with@tikz@firstofone
855 \def\MT@tikz@setup{%
856   \def\MT@maybe@gobble@with@tikz{%
857     \ifnum\tikz@expandcount>\z@
858       \expandafter\@gobble
859     \else
860       \expandafter\@firstofone
861     \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`.)

```

862 \def\MT@setupfont@hook{%

```

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

```

863 \MT@if@false
864 \MT@with@babel@and@T{spanish} \MT@if@true
865 \MT@with@babel@and@T{galician}\MT@if@true
866 \MT@with@babel@and@T{mexican} \MT@if@true
867 \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.)

```

868 \MT@with@package@T{csquotes}{%
869   \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`.

```

870 \MT@if@false
871 \MT@with@package@T{hyperref} \MT@if@true
872 \MT@with@package@T{tex4ht} \MT@if@true
873 \MT@with@package@T{mathastext}\MT@if@true
874 \ifMT@if@\MT@restore@p@h\fi
875 \MT@with@package@T{tikz}\MT@tikz@setup
876 }

```

Check again at the end of the preamble.

```

877 </package>
878 \MT@addto@setup{%
879 <*package>

```

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

```

880 \MT@with@package@T{pdfcpot}{%
881   \MT@error{Detected the 'pdfcpot' package!\MessageBreak
882     '\MT@MT' and 'pdfcpot' may not be used together}{%
883 The 'pdfcpot' package provides an interface to character protrusion.\MessageBreak
884 So does the '\MT@MT' package. Using both packages at the same\MessageBreak
885 time will almost certainly lead to undesired results. Have your choice!}%

```

```

886 }%
887 \MT@with@package@T {ledmac}\MT@ledmac@setup
888 \MT@with@package@T {eledmac}\MT@ledmac@setup
889 \MT@with@package@T {reledmac}\MT@ledmac@setup
890 \MT@with@package@T {xunicode}\MT@xunicodetrue
891 \MT@with@package@T {fontspec}\MT@fontspectrue

```

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

```

892 \MT@gllet\MT@setupfont@hook\@empty
893 \MT@if@false
894 \MT@with@babel@and@T{spanish} \MT@if@true
895 \MT@with@babel@and@T{galician}\MT@if@true
896 \MT@with@babel@and@T{mexican} \MT@if@true
897 \ifMT@if@
898 \g@addto@macro\MT@setupfont@hook{%
899 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
900 \fi
901 \MT@with@package@T{csquotes}{%
902 \ifpackage@later{csquotes}{2005/05/11}{%
903 \g@addto@macro\MT@setupfont@hook\@disablequotes
904 }{%
905 \MT@warning@nl{%
906 Should you receive warnings about unknown slot\MessageBreak
907 numbers, try upgrading the `csquotes' package}%
908 }%
909 }%

```

We disable microtype's additions inside hyperref's \pdfstringdef, which redefines lots of commands. hyperref doesn't work with plain T<sub>E</sub>X, so in that case we don't bother.

```

910 \MT@if@false
911 </package>
912 <plain> \MT@requires@latex2{
913 \MT@with@package@T{hyperref}{%
914 \pdfstringdefDisableCommands{%
915 <*package>
916 \MT@ltx@pickupfont
917 \let\textmicrotypecontext\@secondoftwo
918 \let\microtypecontext\@gobble
919 </package>
920 \def\lststyle{\pdfstringdefWarn\lststyle}%
921 \def\textls#1#\pdfstringdefWarn\textls}%
922 }%
923 <package> \MT@if@true
924 }%
925 <plain> }\relax
926 <*package>
927 \MT@with@package@T{tex4ht}\MT@if@true
928 \MT@with@package@T{mathastext}\MT@if@true
929 \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The listings package makes numbers and letters active,

```

930 \MT@with@package@T{listings}{%
931 \g@addto@macro\MT@cfg@catcodes{%
932 \MT@while@num{"30}{ "3A}{\catcode\@tempcnta=12\relax}%
933 \MT@while@num{"41}{ "5B}{\catcode\@tempcnta=11\relax}%
934 \MT@while@num{"61}{ "7B}{\catcode\@tempcnta=11\relax}%
935 }%

```

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

```

936 \g@addto@macro\MT@setupfont@hook{%
937 \catcode`\=\z@

```

Inside a listing, \space is redefined.

```

938 \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.

```
939 \let\lst@ProcessLetter\@empty
940 }%
941 }%
```

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  $\text{T}_{\text{E}}\text{X}$ , `soul` doesn't register itself the  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  way, so we just test for its main command.

```
942 </package>
943 \ifx\SOU@S@L@@undefined\else
944 \soulregister\lsstyle 0%
945 \soulregister\textls 1%
946 \ifx\XeTeXrevision\undefined
947 \let\MT@SOU@doword\SOU@doword
948 \def\SOU@doword{\pdfadjustspacing=\z@ \MT@SOU@doword}%
949 \fi
950 \fi
951 <*package>
952 \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.

```
953 \MT@with@package@T{pinyin}{%
954 \let\MT@orig@py@macron\py@macron
955 \ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
956 \def\py@macron#1#2{%
957 \MT@ltx@pickupfont
958 \MT@orig@py@macron{#1}{#2}%
959 \MT@MT@pickupfont}%
960 }{%
961 \def\py@macron#1{%
962 \MT@ltx@pickupfont
963 \MT@orig@py@macron{#1}%
964 \MT@MT@pickupfont}%
965 }%
966 }%
967 </package>
968 }
969 </package|letterspace>
```

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

```
970 <package>\expandafter\ifx\the\font\nullfont\normalfont\fi
```

## 14.2 Font setup

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

```
971 <*pdfTeX-def|xetex-def|luatex-def>
972 \def\MT@setupfont{%
```

With  $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$  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.

```
973 <xetex-def|luatex-def> \MT@font
```

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



```
974 \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).

```
975 <pdfTeX-def>\MT@requires@pdfTeX7{
976 <pdfTeX-def|luatex-def>\g@addto@macro\MT@setupfont\MT@copy@font
977 <pdfTeX-def>}\relax
```

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

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

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

```
980 \MT@exp@one@n\MT@find@file\MT@family
981 \ifx\MT@familyalias\@empty \else
982 \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.)

```
983 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
984 }
```

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

```
985 <pdfTeX-def>\MT@requires@pdfTeX6
986 <luatex-def>\MT@requires@luatex3
987 <pdfTeX-def|luatex-def> {\g@addto@macro\MT@setupfont\MT@tracking}\relax
988 \g@addto@macro\MT@setupfont{%
989 \MT@check@font
990 \ifMT@inlist@
991 <debug>\MT@show@pdfannot2%
992 \else
993 \MT@vinfo{Setting up font `\'MT@font'\on@line}%
994 \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.

```
995 \MT@protrusion
996 <pdfTeX-def|luatex-def> \MT@expansion
997 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
998 <*pdfTeX-def>
999 \MT@requires@pdfTeX6{
1000 \g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}
1001 }\relax
1002 </pdfTeX-def>
```

Disable ligatures (pdfTeX 1.30).

```
1003 <pdfTeX-def>\MT@requires@pdfTeX5{
1004 <pdfTeX-def|luatex-def>\g@addto@macro\MT@setupfont\MT@no@ligatures
1005 <pdfTeX-def>}\relax
1006 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
1007 <debug>\MT@show@pdfannot1%
```

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

```
1008 \MT@register@font
1009 \fi
1010 }
1011 </pdfTeX-def|xetex-def|luatex-def>
```

\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.

```
1012 <*pdfTeX-def|luatex-def>
1013 \let\MT@copy@font\relax
1014 <pdfTeX-def>\MT@requires@pdfTeX7{
1015 \def\MT@copy@font@{%
```

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

```
1016 \xdef\MT@font@copy{\csname\MT@font/\MT@pr@context/\MT@ex@context\endcsname}%
1017 \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.

```
1018 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
1019 \expandafter\ifx\MT@font@orig\relax
1020 \MT@exp@two@c\MT@glet\MT@font@orig\font@name
1021 \else
1022 \MT@exp@two@c\let\font@name\MT@font@orig
1023 \fi
1024 <pdfTeX-def> \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.

```
1025 <luatex-def> \MT@exp@two@c\MT@lua@copyfont\meaning\font@name\@nil
1026 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%
```

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

```
1027 \MT@map@clist@c\MT@active@features{%
1028 \MT@exp@cs\ifx\MT@\@nameuse\MT@abbr@#1}\relax\else
1029 \def\@tempa{#1}%
1030 \MT@exp@cs\MT@map@tlist@c\MT@##1@doc@contexts\MT@rem@from@list
1031 \fi
1032 }%
1033 \fi
1034 \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

```
1035 \let\font@name\MT@font@copy
```

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

```
1036 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
1037 }
```

\MT@rem@from@list

```
1038 \def\MT@rem@from@list#1{%
1039 \MT@exp@cs\ifx\MT@\@tempa @#1font@list}\relax\else
1040 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
1041 \MT@font \csname \MT@\@tempa @#1font@list\endcsname
1042 \fi
1043 }
1044 <pdfTeX-def>\relax
```

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

```

1045 <luatex-def>\def\MT@lua@copyfont #1 #2 #3\@nil{%
1046 <luatex-def> \global\expandafter\font\MT@font@copy=#3\relax}
1047 <pdfTeX-def|luatex-def>

```

*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`    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 – only tracking still doesn’t work (it seems that `\letterspacefont` uses the `\fontdimen 6` from the original font). X<sub>Y</sub>TeX doesn’t provide an equivalent to `\pdffontsize`, so we use the nominal size instead.

`\MT@dimen@six`

```

1048 <*pdfTeX-def|luatex-def|xetex-def>
1049 \def\MT@fix@fontdimen@six{%
1050   \ifnum\fontdimen6\MT@font=\z@
1051     \fontdimen6\MT@font=%
1052 <pdfTeX-def> \pdffontsize\MT@font
1053 <luatex-def> \MT@requires@luatex4{\pdffeedback fontsize}{\pdffontsize}\MT@font
1054 <xetex-def> \MT@size pt
1055   \MT@info{Fixing zero \string\fontdimen 6 for font ‘\MT@font’\MessageBreak
1056     (new value: \the\fontdimen6\MT@font)}%
1057   \MT@glet@nc{\MT@font-fake6}\@empty
1058   \fi
1059   \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1060 }
1061 </pdfTeX-def|luatex-def|xetex-def>

```

`\MT@split@name`    Split up the font name (`<#6>` may be a protrusion/expansion context and/or a letterspacing amount). With `fontspec` we also need to remove its internal instance counter.

`\MT@encoding`

`\MT@family`

`\MT@series` 1062 `<*package>`

`\MT@shape` 1063 `\def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%`

`\MT@size` 1064 `\def\MT@encoding{#1}%`

1065 `\ifMT@fontspec`

```

1066 \edef\MT@family{\MT@scrubfeature#2()\relax}%
1067 \else
1068 \def\MT@family{#2}%
1069 \fi
1070 \def\MT@series {#3}%
1071 \def\MT@shape {#4}%
1072 \def\MT@size {#5}%
1073 \MT@fix@fontdimen@six

\MT@familyalias Alias family?

1074 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
1075 {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
1076 {\let\MT@familyalias\@empty}%
1077 }

\MT@scrubfeature Remove one resp. all feature counters (fontspec).
\MT@scrubfeatures
1078 \def\MT@scrubfeature#1(#2)#3\relax{#1}
1079 \def\MT@scrubfeatures#1(#2)#3\relax{%
1080 #1%
1081 \ifx\relax#3\relax\else
1082 \MT@scrubfeatures#3\relax
1083 \fi
1084 }

\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
1085 \newif\ifMT@do
1086 \def\MT@maybe@do#1{%
    (but only if the feature isn't globally set to false)
1087 \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).

1088 \MT@dotrue
1089 \edef\@tempa{\csname MT@#1@setname\endcsname}%
1090 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1091 \MT@ifdefined@n@TF{MT@checklist@#1}%
1092 {\csname MT@checklist@#1\endcsname}%
1093 {\MT@checklist@{#1}}}%
1094 {#1}%
1095 }%
1096 \else
1097 \MT@dofalse
1098 \fi
1099 \ifMT@do

    \MT@feat stores the current feature.

1100 \def\MT@feat{#1}%
1101 \csname MT@set@#1@codes\endcsname
1102 \else
1103 \MT@ifstreq{#1}{tr}%
1104 {\let\MT@info@nottracking\MT@info@nottracking@}%
1105 {\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}%
1106 \fi
1107 }

\MT@info@nottracking To defer the message to after the font has actually been logged.
\MT@info@nottracking@
1108 \let\MT@info@nottracking\relax
1109 \def\MT@info@nottracking@{\MT@vinfo{... No tracking}}

\MT@dinfo@list
1110 <debug>\def\MT@dinfo@list#1#2#3{\MT@dinfo@n1{1}{\@nameuse{MT@abbr@#1}: #2

```

```
1111 <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).

```
1112 \def\MT@checklist@#1#2{%
1113 <!debug> \MT@ifdefined@n@T
1114 <debug> \MT@ifdefined@n@TF
1115 {MT@#2list@#1@ \@tempa}{%}
```

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

```
1116 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
1117 \csname MT@#1\expandafter\endcsname
1118 \csname MT@#2list@#1@ \@tempa\endcsname
1119 \ifMT@inlist@
1120 <debug>\MT@info@list{#2}{#1}{in}%
1121 \MT@dotrue
1122 \else
1123 <debug>\MT@info@list{#2}{#1}{not in}%
1124 \MT@dofalse
1125 \expandafter\MT@clist@break
1126 \fi
1127 }%
```

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.

```
1128 <debug> {\MT@info@list{#2}{#1}{}}%
1129 }
```

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

```
1130 \def\MT@checklist@family#1{%
1131 <!debug> \MT@ifdefined@n@T
1132 <debug> \MT@ifdefined@n@TF
1133 {MT@#1list@family@ \@tempa}{%
1134 \MT@exp@two@n\MT@in@clist
1135 \MT@family{\csname MT@#1list@family@ \@tempa\endcsname}%
1136 \ifMT@inlist@
1137 <debug>\MT@info@list{#1}{family}{in}%
1138 \MT@dotrue
1139 \else
1140 <debug>\MT@info@list{#1}{family}{not in}%
1141 \MT@dofalse
1142 \ifx\MT@familyalias\empty \else
1143 \MT@exp@two@n\MT@in@clist
1144 \MT@familyalias{\csname MT@#1list@family@ \@tempa\endcsname}%
1145 \ifMT@inlist@
1146 <debug> \MT@info@list{#1}{family alias}{in}%
1147 \MT@dotrue
1148 <debug>\else\MT@info@list{#1}{family alias}{not in}%
1149 \fi
1150 \fi
1151 \fi
1152 \ifMT@do \else
1153 \expandafter\MT@clist@break
1154 \fi
1155 }%
1156 <debug> {\MT@info@list{#1}{family}{}}%
1157 }
```

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

```
1158 \def\MT@checklist@size#1{%
1159 <!debug> \MT@ifdefined@n@T
1160 <debug> \MT@ifdefined@n@TF
1161 {MT@#1list@size@ \@tempa}{%
1162 \MT@exp@cs\MT@in@rlist{MT@#1list@size@ \@tempa}%}
```

```

1163   \ifMT@inlist@
1164   <debug>\MT@info@list{#1}{size}{in}%
1165   \MT@dotrue
1166   \else
1167   <debug>\MT@info@list{#1}{size}{not in}%
1168   \MT@dofalse
1169   \expandafter\MT@clist@break
1170   \fi
1171   }%
1172   <debug> {\MT@info@list{#1}{size}}}%
1173   }

```

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

```

1174 \def\MT@checklist@font#1{%
1175 <!debug> \MT@ifdefined@n@T
1176 <debug> \MT@ifdefined@n@TF
1177   {MT@#1list@font@%tempa}%

```

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

```

1178   \edef@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1179   \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1180   \@tempb \csname MT@#1list@font@%tempa\endcsname
1181   \ifMT@inlist@
1182   <debug>\MT@info@list{#1}{font}{in}%
1183   \expandafter\MT@clist@break
1184   \else
1185   <debug>\MT@info@list{#1}{font}{not in}%
1186   \MT@dofalse
1187   \fi
1188   }%
1189   <debug> {\MT@info@list{#1}{font}}}%
1190   }

```

### 14.2.1 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.

```

1191 \newif\ifMT@nofamily
1192 </package>

```

\MT@protrusion     Set up for protrusion?

```

1193 <*pdfTeX-def|xetex-def|luatex-def>
1194 \def\MT@protrusion{\MT@maybe@do{pr}}

```

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

```

1195 \def\MT@set@pr@codes{%
1196   \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).

```

1197   \MT@if@list@exists{%
1198   \ifMT@nofamily
1199   \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
1200   \MT@info@n{Loading generic protrusion settings for font family\MessageBreak
1201   ~\MT@family' (encoding: \MT@encoding).\MessageBreak
1202   For optimal results, create family-specific settings.\MessageBreak
1203   See the microtype manual for details}%
1204   \MT@gl@et@nc{\MT@encoding-\MT@family-settings}\@empty
1205   }%
1206   \fi

```

```

1207 \MT@get@opt
1208 \MT@reset@pr@codes

Get the name of the inheritance list and parse it.
1209 \MT@get@inh@list

Set an input encoding?
1210 \MT@set@inputenc{c}%

Load additional lists?
1211 \MT@load@list\MT@pr@c@name
1212 \MT@set@listname

Load the main list.
1213 \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
1214 \expandafter\MT@set@codes\@tempc,\relax,%
1215 }\MT@reset@pr@codes
1216 }

\MT@set@all@pr Set all protrusion codes of the font.
1217 \def\MT@set@all@pr#1#2{%
1218 debug\MT@info{n1}{3}{-- lp/rp: setting all to #1/#2}%
1219 \let\MT@temp\@empty
1220 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1}}%
1221 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2}}%
1222 \MT@do@font\MT@temp
1223 }

\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.
1224 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
1225 \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.
1226 \def\MT@the@pr@code{\@tempcntb}
1227 pdftex-def|luatex-def
1228 pdftex-def\MT@requirespdftex6
1229 luatex-def\MT@requiresluatex3
1230 {\def\MT@the@pr@code@tr{%
1231 \numexpr\@tempcntb+\MT@letterspace@/2\relax
1232 }
1233 }\relax
1234 pdftex-def|luatex-def

\MT@set@codes Split up the values and set the codes.
1235 \def\MT@set@codes#1,{%
1236 \ifx\relax#1\@empty\else
1237 \MT@split@codes #1==\relax
1238 \expandafter\MT@set@codes
1239 \fi
1240 }

\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.
1241 \def\MT@split@codes#1=#2=#3\relax{%
1242 \def\@tempa{#1}%
1243 \ifx\@tempa\@empty \else
1244 \MT@get@slot
1245 pdftex-def|luatex-def \ifnum\MT@char > \m@ne
1246 xetex-def \ifx\MT@char\@empty \else

```

```

1247 \MT@get@char@unit
1248 \csname MT@MT@feat @split@val\endcsname#2\relax
1249 \fi
1250 \fi
1251 }

\MT@pr@split@val
1252 \def\MT@pr@split@val#1,#2\relax{%
1253 \def\@tempb{#1}%
1254 \MT@ifempty\@tempb\relax{%
1255 \MT@scale@to@em
1256 \lpcode\MT@font\MT@char=\MT@the@pr@code
1257 <debug>\MT@info@n1{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char\space: [#1]}%
1258 }%
1259 \def\@tempb{#2}%
1260 \MT@ifempty\@tempb\relax{%
1261 \MT@scale@to@em
1262 \rprcode\MT@font\MT@char=\MT@the@pr@code
1263 <debug>\MT@info@n1{4}{;;; rp (\MT@char): \number\rprcode\MT@font\MT@char\space: [#2]}%
1264 }%

```

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

```

1265 \MT@ifdefined@c@T\MT@pr@inh@name{%
1266 \MT@ifdefined@n@T\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1267 \MT@exp@cs\MT@map@tlist@c
1268 {MT@inh@\MT@pr@inh@name @\MT@char @}%
1269 \MT@set@pr@heirs
1270 }%
1271 }%
1272 }

```

`\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. `\rprcode`, since this would disallow protrusion factors larger than the character width (since `\lpcode`’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`.

```

1273 <pdfTeX-def>\MT@requires@pdfTeX3{
1274 \def\MT@scale@to@em{%
1275 \@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`.

```

1276 \MT@scale\@tempcntb \@tempb \MT@dimen@six
1277 \ifnum\@tempcntb=z@ \else
1278 \MT@scale@factor
1279 \fi
1280 }

```

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

```

1281 \def\MT@get@charwd{%
1282 <*pdfTeX-def>
1283 ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1284 ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%

```



```

1285 ^^Q \MT@count=\wd\z@
1286 </pdfTeX-def>
1287 <luatex-def> \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).
1288 <*xetex-def>
1289 \ifnum\MT@char@<\z@
1290   \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}%
1291   \MT@count=\wd\z@
1292 \else
1293   \MT@count=\fontcharwd\MT@font\MT@char@\relax
1294 \fi
1295 </xetex-def>
1296 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1297 }

```

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.

```

1298 <*pdfTeX-def>
1299 \MT@requires@pdfTeX6{
1300   \g@addto@macro\MT@get@charwd{%
1301     \MT@ifdefined@c@T\MT@letterspace@
1302     {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1303   }
1304 }\relax
1305 }

```

No adjustment with versions 0.14f and 0.14g.

```

1306 \def\MT@scale@to@em{%
1307   \MT@count=\@tempb\relax
1308   \ifnum\MT@count=\z@ \else
1309     \MT@scale@factor
1310   \fi
1311 }

```

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

```

1312 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1313 }
1314 </pdfTeX-def>
1315 </pdfTeX-def|xetex-def|luatex-def>

```

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

```

1316 <*package>
1317 \def\MT@get@font@dimen#1{%
1318   \ifnum\fontdimen#1\MT@font=\z@
1319     \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1320       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1321       You should use a different 'unit' for \MT@curr@list@name}%
1322   \else
1323     \MT@count=\fontdimen#1\MT@font
1324   \fi
1325 }

```

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

```

1326 \def\MT@info@missing@char{%
1327   \MT@info@n1{Character '\the\MT@toks'
1328     ^^X \ifnum\MT@char@<\z@ is missing\else
1329     ^^X \iffontchar\MT@font\MT@char@
1330       has a width of 0pt
1331     ^^X \else is missing\fi\fi
1332     ^^Q \MessageBreak (it's probably missing)
1333     \MessageBreak in font '\MT@font'. \MessageBreak

```

```
1334 Ignoring protrusion settings for this character}%
1335 }
```

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

```
1336 \def\MT@scale@factor{%
1337   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1338     \expandafter\MT@scale\expandafter \@tempcntb
1339     \csname MT@\MT@feat @factor@\endcsname \@m
1340   \fi
1341   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1342     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1343   \else
1344     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1345       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1346     \fi
1347   \fi
1348 }
```

`\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.

```
1349 \def\MT@warn@code@too@large#1{%
1350   \@tempcnta=#1\relax
1351   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1352     \expandafter\MT@scale\expandafter \@tempcnta\expandafter
1353     \@m \csname MT@\MT@feat @factor@\endcsname
1354   \fi
1355   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1356   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@temp\space
1357     is too large for character\MessageBreak
1358     `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1359     Setting it to the maximum of \number\@tempcnta}%
1360   \@tempcntb=#1\relax
1361 }
```

`\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.).

```
1362 \def\MT@get@opt{%
1363   \MT@set@listname
```

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

```
\MT@sp@factor@ 1364 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1365   \MT@let@nn{MT@\MT@feat @factor@}
1366   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1367   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1368     \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1369   }{%
1370   \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1371   }%
```

`\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@ 1372 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1373   \MT@let@nn{MT@\MT@feat @unit@}%
1374   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1375   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1376   \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1377     relative to character widths}%
1378   \else
1379     \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1380     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1381       relative to width of space}%
1382   }
```

```

1382     \fi
1383     \fi
1384   }%
1385   \MT@let@nn{MT@MT@feat @unit@}{MT@MT@feat @unit}%
1386 }%

```

\MT@get@space@unit      The codes are either relative to character widths, or to a fixed width. For spacing 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.

```

1387   \let\MT@get@char@unit\relax
1388   \let\MT@get@space@unit\@gobble
1389   \MT@exp@cs\ifx{MT@MT@feat @unit@}\@empty
1390     \let\MT@get@char@unit\MT@get@charwd
1391   \else
1392     \MT@exp@cs\ifx{MT@MT@feat @unit@}\m@ne
1393     \let\MT@get@space@unit\MT@get@font@dimen
1394   \else
1395     \MT@exp@cs\MT@get@unit{MT@MT@feat @unit}%
1396   \fi
1397 \fi

```

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

```

1398   \MT@ifdefined@n@T{MT@MT@feat @c@csname MT@MT@feat @c@name\endcsname @preset}{%
1399     \csname MT@preset@MT@feat\endcsname
1400     \MT@let@nc{MT@reset@MT@feat @codes}\relax
1401   }%
1402 }

```

\MT@get@unit      If unit contains an em or ex, we use the corresponding \fontdimen to obtain the 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.

```

1403 \def\MT@get@unit#1{%
1404   \expandafter\MT@get@unit@#1 e!\@nil
1405   \ifx\x\@empty\else\let#1\x\fi
1406   \@defaultunits\@tempdima#1 pt\relax\@nnil
1407   \ifdim\@tempdima=\z@
1408     \MT@warning@n1{%
1409       Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1410       width. Setting factors of list \@nameuse{MT@MT@feat @c@name}'\MessageBreak
1411       relative to character widths instead}%
1412     \let#1\@empty
1413     \let\MT@get@char@unit\MT@get@charwd
1414   \else
1415     \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1416       to \the\@tempdima}%
1417     \MT@count=\@tempdima\relax
1418   \fi
1419 }
1420 \def\MT@get@unit@#1e#2#3\@nil{%
1421   \ifx\#3\@empty\let\x\@empty\else
1422     \if m#2%
1423       \edef\x{#1\fontdimen6\MT@font}%
1424     \else
1425       \if x#2%
1426         \edef\x{#1\fontdimen5\MT@font}%
1427       \fi
1428     \fi
1429   \fi
1430 }

```

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

```

1431 \def\MT@set@inputenc#1{%

```

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

```
1432 \def\MT@cat{#1}%
1433 \edef\@tempa{\MT@feat @#1\csname MT@\MT@feat @#1\name\endcsname @inputenc}%
1434 \MT@ifdefined{n@T\@tempa\MT@set@inputenc@
1435 }
```

`\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.

```
1436 \MT@addto@setup{%
1437 \ifpackageloaded{inputenc}{%
1438 \ifpackageafter{inputenc}{2006/02/22}{%
1439 \def\MT@set@inputenc{%
1440 \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1441 \MT@load@inputenc
1442 }%
1443 }%
1444 \let\MT@set@inputenc\MT@load@inputenc
1445 }%
1446 }%
1447 \def\MT@set@inputenc{%
1448 \MT@warning@n1{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1449 \MessageBreak package isn't loaded. Ignoring input encoding}%
1450 }%
1451 }%
1452 }
```

`\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.

```
1453 \def\MT@load@inputenc{%
1454 \MT@cfg@catcodes
1455 <debug>\MT@info@n1{1}{loading input encoding: \@nameuse{\@tempa}}%
1456 \inputencoding{\@nameuse{\@tempa}}%
1457 }
1458 </package>
```

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

```
1459 <*pdfTeX-def|xetex-def|luatex-def>
1460 \def\MT@set@pr@heirs#1{%
1461 \lcode\MT@font #1=\lcode\MT@font\MT@char\relax
1462 \rcode\MT@font #1=\rcode\MT@font\MT@char\relax
1463 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1464 <debug>\MT@info@n1{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char\space/%
1465 \number\rcode\MT@font\MT@char\space}%
1466 }
```

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

```
\MT@preset@pr@
1467 \def\MT@preset@pr{%
1468 \expandafter\expandafter\expandafter\MT@preset@pr@
1469 \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1470 }
1471 \def\MT@preset@pr@#1,#2\@nil{%
1472 \ifx\MT@pr@unit@\empty
1473 \MT@warn@preset@twidth{pr}%
1474 \let\MT@preset@aux\MT@preset@aux@factor
1475 \else
1476 \def\MT@preset@aux{\MT@preset@aux@space2}%
1477 \fi
1478 \MT@ifempty{#1}{\let\@tempa\empty}{\MT@preset@aux{#1}\@tempa}%
1479 \MT@ifempty{#2}{\let\@tempb\empty}{\MT@preset@aux{#2}\@tempb}%
1480 \MT@set@all@pr\@tempa\@tempb
1481 }
```

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

`\MT@preset@aux@factor`

`\MT@preset@aux@space`

```

1482 \def\MT@preset@aux@factor#1#2{%
1483   \@tempcntb=#1\relax
1484   \MT@scale@factor
1485   \edef#2{\number\@tempcntb}%
1486 }
1487 \def\MT@preset@aux@space#1#2#3{%
1488   \def\@tempb{#2}%
1489   \MT@get@space@unit#1%
1490   \MT@scale@to@em
1491   \edef#3{\number\@tempcntb}%
1492 }

\MT@warn@preset@tewidth

1493 \def\MT@warn@preset@tewidth#1{%
1494   \MT@warning@nl{%
1495     Cannot preset characters relative to their widths\MessageBreak
1496     for \@nameuse{MT@abbr#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1497     \MessageBreak relative to lem instead}%
1498 }
1499 (pdftex-def|xetex-def|luatex-def)

```

### 14.2.2 Expansion

`\MT@expansion`      Set up for expansion?

```

1500 (pdftex-def|luatex-def)
1501 \def\MT@expansion{\MT@maybe@do{ex}}

```

`\MT@set@ex@codes@`      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).

```

1502 \def\MT@set@ex@codes@{%
1503   \MT@if@list@exists{%
1504     \MT@get@ex@opt
1505     \let\MT@get@char@unit\relax
1506     \MT@reset@ef@codes
1507     \MT@get@inh@list
1508     \MT@set@inputenc{c}%
1509     \MT@load@list\MT@ex@c@name
1510     \MT@set@listname
1511     \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
1512     \expandafter\MT@set@codes\@tempc,\relax,%
1513     \MT@expandfont
1514   }\relax
1515 }
1516 (pdftex-def|luatex-def)

```

`\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.

```

1517 (package)\newif\ifMT@nonselected

1518 (pdftex-def|luatex-def)
1519 \def\MT@set@ex@codes@n{%
1520   \MT@nonselectedtrue
1521   \MT@if@list@exists
1522   \MT@get@ex@opt
1523   {%
1524     \let\MT@stretch@ \MT@stretch
1525     \let\MT@shrink@ \MT@shrink
1526     \let\MT@step@ \MT@step

```

```

1527 \let\MT@auto@ \MT@auto
1528 \let\MT@ex@factor@ \MT@ex@factor
1529 }%
1530 \MT@reset@ef@codes
1531 \MT@expandfont
1532 \MT@nonselectedfalse
1533 }

```

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

```
1534 \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.

```

1535 <*luatex-def>
1536 \MT@requires@luatex3{
1537 \MT@requires@luatex4{\let\pdffontexpand\expandglyphsinfont}\relax
1538 \ifnum\luatexversion<79
1539 \def\MT@expandfont{%
1540 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ autoexpand\relax
1541 }
1542 \else
1543 \def\MT@expandfont{%
1544 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@\relax
1545 }
1546 \fi
1547 }{
1548 </luatex-def>
1549 \def\MT@expandfont{%
1550 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1551 }
1552 <luatex-def>}}

```

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

```

1553 \def\MT@set@all@ex#1{%
1554 <debug>\MT@info@n1{3}{-- ex: setting all to \number#1}%
1555 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1556 }
1557 \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).

```

1558 <pdfTeX-def>\MT@requires@pdfTeX4
1559 <luatex-def>\MT@requires@luatex5
1560 {
1561 \def\MT@reset@ef@codes{%
1562 \ifnum\MT@ex@factor@=\@m \else
1563 \MT@reset@ef@codes@
1564 \fi
1565 }
1566 }{
1567 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1568 }

```

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

```

1569 \def\MT@ex@split@val#1\relax{%
1570 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1571 \ifnum\MT@ex@factor@=\@m \else
1572 \MT@scale\@tempcntb \MT@ex@factor@ \@m

```

```

1573 \fi
1574 \ifnum\@tempcntb > \MT@ex@max
1575 \MT@warn@ex@too@large\MT@ex@max
1576 \else
1577 \ifnum\@tempcntb < \MT@ex@min
1578 \MT@warn@ex@too@large\MT@ex@min
1579 \fi
1580 \fi
1581 \efcode\MT@font\MT@char=\@tempcntb
1582 <debug>\MT@info@n1{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1583 \MT@ifdefined@c@T\MT@ex@inh@name{%
1584 \MT@ifdefined@n@T\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1585 \MT@exp@cs\MT@map@tlist@c\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1586 }%
1587 }%
1588 }

```

\MT@warn@ex@too@large

```

1589 \def\MT@warn@ex@too@large#1{%
1590 \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1591 character\MessageBreak ``the\MT@toks' in \MT@curr@list@name.\MessageBreak
1592 Setting it to the maximum of \number#1}%
1593 \@tempcntb=#1\relax
1594 }

```

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

```

\MT@ex@factor@ 1595 \def\MT@get@ex@opt{%
\MT@stretch@ 1596 \MT@set@listname
1597 \MT@ifdefined@n@TF\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1598 \MT@let@cn\MT@ex@factor@\MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1599 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1600 }{%
1601 \let\MT@ex@factor@\MT@ex@factor
1602 }%
1603 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1604 \MT@get@ex@opt@{shrink}{Setting shrink limit to \number\MT@shrink@}%
1605 \MT@get@ex@opt@{step}{Setting expansion step to \number\MT@step@}%
1606 <luatex-def> \MT@requires@luatex3\relax{%
1607 \MT@get@ex@opt@{auto}{\MT@ifstreq{\MT@auto@}{autoexpand}{En}{Dis}abling automatic expansion}%
1608 <luatex-def> }%
1609 \MT@ifdefined@n@T\MT@ex@c@\MT@ex@c@name @preset}{%
1610 \MT@preset@ex
1611 \let\MT@reset@ef@codes\relax
1612 }%
1613 }

```

\MT@get@ex@opt@

```

1614 \def\MT@get@ex@opt@#1#2{%
1615 \MT@ifdefined@n@TF\MT@ex@c@\MT@ex@c@name @#1}{%
1616 \MT@let@nn\MT@#1@{\MT@ex@c@\MT@ex@c@name @#1}%
1617 \MT@vinfo{... : #2}%
1618 }{%
1619 \MT@let@nn\MT@#1@{\MT@#1}%
1620 }%
1621 }

```

\MT@set@ex@heirs

```

1622 \def\MT@set@ex@heirs#1{%
1623 \efcode\MT@font#1=\efcode\MT@font\MT@char
1624 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1625 <debug>\MT@info@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1626 }

```

\MT@preset@ex

```

1627 \def\MT@preset@ex{%
1628   \@tempcntb=\csname MT@ex@cc\MT@ex@cc@name @preset\endcsname\relax
1629   \MT@scale@factor
1630   \MT@set@all@ex\@tempcntb
1631 }
1632 (/pdfTeX-def|luatex-def)

```

### 14.2.3 Interword spacing (glue)

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

```

1633 (*pdfTeX-def)
1634 \MT@requirespdfTeX6{
1635 \def\MT@spacing{\MT@maybe@do{sp}}

```

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

```

1636 \def\MT@set@sp@codes{%
1637   \MT@if@list@exists{%
1638     \MT@get@opt
1639     \MT@reset@sp@codes
1640     \MT@get@inh@list
1641     \MT@set@inputenc{c}%
1642     \MT@load@list\MT@sp@cc@name
1643     \MT@set@list@name
1644     \MT@let@cn\@tempc{MT@sp@cc\MT@sp@cc@name}%
1645     \expandafter\MT@set@codes\@tempc,\relax,%
1646   }\MT@reset@sp@codes
1647 }

```

\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).

```

1648 \def\MT@sp@split@val#1,#2,#3\relax{%
1649   \def\@tempb{#1}%
1650   \MT@ifempty\@tempb\relax{%
1651     \MT@get@space@unit2%
1652     \MT@scale@to@em
1653     \knbscode\MT@font\MT@char=\@tempcntb
1654 (debug)\MT@info@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1655   }%
1656   \def\@tempb{#2}%
1657   \MT@ifempty\@tempb\relax{%
1658     \MT@get@space@unit3%
1659     \MT@scale@to@em
1660     \stbscode\MT@font\MT@char=\@tempcntb
1661 (debug)\MT@info@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1662   }%
1663   \def\@tempb{#3}%
1664   \MT@ifempty\@tempb\relax{%
1665     \MT@get@space@unit4%
1666     \MT@scale@to@em
1667     \shbscode\MT@font\MT@char=\@tempcntb
1668 (debug)\MT@info@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1669   }%
1670   \MT@ifdefined@c@T\MT@sp@inh@name{%
1671     \MT@ifdefined@nT{MT@inh\MT@sp@inh@name @\MT@char @}{%
1672       \MT@exp@cs\MT@map@tlist@c{MT@inh\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
1673     }%
1674   }%
1675 }

```

\MT@set@sp@heirs

```

1676 \def\MT@set@sp@heirs#1{%
1677   \knbscode\MT@font#1=\knbscode\MT@font\MT@char

```



```

1678 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1679 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1680 (debug)\MT@info{n1}{2}{-- heir of \MT@char: #1}%
1681 (debug)\MT@info{n1}{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1682 (debug) \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1683 }

\MT@set@all@sp
\MT@reset@sp@codes 1684 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1685 (debug)\MT@info{n1}{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1686 \let\MT@temp\@empty
1687 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1688 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1689 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1690 \MT@do@font\MT@temp
1691 }
1692 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1693 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1694 \def\MT@preset@sp{%
1695 \expandafter\expandafter\expandafter\MT@preset@sp@
1696 \cscname \MT@sp@cc@MT@sp@cc@name @preset\endcsname\@nil
1697 }
1698 \def\MT@preset@sp@#1,#2,#3\@nil{%
1699 \ifx\MT@sp@unit\@empty
1700 \MT@warn@preset@tewidth{sp}%
1701 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1702 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1703 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1704 \else
1705 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1706 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1707 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1708 \fi
1709 \MT@set@all@sp\@tempa\@tempc\@tempb
1710 }
1711 }\relax

```

#### 14.2.4 Additional kerning

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

```

1712 \MT@requires@pdftex6{
1713 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1714 \def\MT@set@kn@codes{%
1715 \MT@if@list@exists{%
1716 \MT@get@opt
1717 \MT@reset@kn@codes
1718 \MT@get@inh@list
1719 \MT@set@inputenc{c}%
1720 \MT@load@list\MT@kn@cc@name
1721 \MT@set@listname
1722 \MT@let@cn\@tempc{MT@kn@cc@MT@kn@cc@name}%
1723 \expandafter\MT@set@codes\@tempc,\relax,%
1724 }\MT@reset@kn@codes
1725 }

```

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

```

1726 \def\MT@kn@split@val#1,#2\relax{%
1727 \def\@tempb{#1}%
1728 \MT@ifempty\@tempb\relax{%

```

```

1729 \MT@get@space@unit2%
1730 \MT@scale@to@em
1731 \knbcode\MT@font\MT@char=\@tempcntb
1732 <debug>\MT@info@n1{4}{;;; knbc (\MT@char): \number\knbcode\MT@font\MT@char: [#1]}%
1733 }%
1734 \def\@tempb{#2}%
1735 \MT@ifempty\@tempb\relax%
1736 \MT@get@space@unit2%
1737 \MT@scale@to@em
1738 \knacode\MT@font\MT@char=\@tempcntb
1739 <debug>\MT@info@n1{4}{;;; knac (\MT@char): \number\knacode\MT@font\MT@char: [#2]}%
1740 }%
1741 \MT@ifdefined@c@T\MT@kn@inh@name{%
1742 \MT@ifdefined@nT\MT@inh@MT@kn@inh@name @\MT@char @}%
1743 \MT@exp@cs\MT@map@tlist@c\MT@inh@MT@kn@inh@name @\MT@char @\MT@set@kn@heirs
1744 }%
1745 }%
1746 }

\MT@set@kn@heirs

1747 \def\MT@set@kn@heirs#1{%
1748 \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1749 \knacode\MT@font#1=\knacode\MT@font\MT@char
1750 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1751 <debug>\MT@info@n1{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1752 <debug> \number\knacode\MT@font\MT@char}%
1753 }

\MT@set@all@kn

\MT@reset@kn@codes 1754 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1755 <debug>\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1756 \let\MT@temp\@empty
1757 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1758 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knacode\MT@font\@tempcnta=#2\relax}}%
1759 \MT@do@font\MT@temp
1760 }
1761 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1762 \let\MT@reset@kn@codes\relax

\MT@preset@kn

\MT@preset@kn@ 1763 \def\MT@preset@kn{%
1764 \expandafter\expandafter\expandafter\MT@preset@kn@
1765 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1766 }
1767 \def\MT@preset@kn@#1,#2\@nil{%
1768 \ifx\MT@kn@unit@\@empty
1769 \MT@warn@preset@twidth{kn}%
1770 \let\MT@preset@aux\MT@preset@aux@factor
1771 \else
1772 \def\MT@preset@aux{\MT@preset@aux@space2}%
1773 \fi
1774 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1775 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1776 \MT@set@all@kn\@tempa\@tempb
1777 }
1778 }\relax
1779 </pdfTeX-def>

```

### 14.2.5 Tracking

This only works with pdfTeX 1.40 or LuaTeX 0.62.

```

1780 <*pdfTeX-def|luatex-def>
1781 <pdfTeX-def>\MT@requires@pdfTeX6
1782 <luatex-def>\MT@requires@luatex3

```

```

1783 {
\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 1784 \let\MT@tr@font@list\empty
1785 \def\MT@tracking@{%
1786   \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1787   \ifMT@inlist\else
1788     \MT@maybe@do{tr}%
1789     \ifMT@do\else
1790       \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1791     \fi
1792   \fi
1793 }
1794 \pdfTeX-def|luatex-def
1795 \pdfTeX-def|luatex-def|letterspace\let\MT@tracking
1796 \pdfTeX-def|luatex-def\MT@tracking@
1797 \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 if the original font's \fontdimen 6 is zero, in which case
                    we issue a warning (once for every font).
1798 \pdfTeX-def|luatex-def|letterspace
1799 \def\MT@set@tr@codes{%
1800 \pdfTeX-def|luatex-def
1801   \MT@vinfo{Tracking font ~\MT@font'\on@line}%
1802   \MT@ifdefined@n@TF{\MT@font-fake6}{%
1803     \expandafter\ifx\csname\MT@font-fake6\endcsname\empty
1804       \MT@warning@n1{%
1805         Font ~\MT@font' does not specify its\MessageBreak
1806         \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
1807         \nameuse{MT@abbr@MT@feat} will not work with this font}%
1808       \MT@gl@et@nc{\MT@font-fake6}\relax
1809     \fi
1810   }{%
1811     \MT@if@list@exists
1812     \MT@get@tr@opt
1813     \relax
1814 \pdfTeX-def|luatex-def
1815   \MT@ifdefined@c@TF\MT@letterspace@ \relax{\let\MT@letterspace@ \MT@letterspace}%
1816   \ifnum\MT@letterspace@=\z@

Zero tracking requires special treatment.
1817   \MT@set@tr@zero
1818   \else
1819 \pdfTeX-def|luatex-def\MT@vinfo{... Tracking by \number\MT@letterspace@}%

Letterspacing only works in PDF mode.
1820   \MT@warn@tracking@DVI

\MT@lsfont    The letterspaced font instances are saved in macros \font name)/letterspacing
              amount)ls.
              In contrast to \MT@font, which may reflect the font characteristics more accurately
              (taking substitutions into account), \font@name is guaranteed to correspond
              to an actual font identifier.
1821   \xdef\MT@lsfont{\csname\expandafter\string\font@name
1822     \number\MT@letterspace@ls\endcsname}%
1823   \expandafter\ifx\MT@lsfont\relax
1824 \debug\MT@edinfo@n1{1}{... new letterspacing instance}%

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

```

font again.

```
1825 \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.

```
1826 <*luatex-def|letterspace>
1827 \MT@if@luaotf@font{%
1828 <luatex-def&debug>\MT@info@n1{1}{... luaotf font: \MessageBreak
1829 <luatex-def&debug> \expandafter\fontname\font@name}%
1830 \ifnum\MT@letterspace@<z@\def\MT@minus{-}\else\let\MT@minus\empty\fi
1831 \global\expandafter\font\MT@lsfont=%
1832 \expandafter\MT@exp@two@c\expandafter\MT@ls@fontspec@font
1833 \expandafter\fontname\expandafter\font@name\space \@nil
1834 }{%
1835 </luatex-def|letterspace>
1836 <luatex-def&debug>\MT@info@n1{1}{... legacy font}%
1837 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
1838 <luatex-def|letterspace> }%
```

Scale interword spacing (not configurable in letterspace).

```
1839 <*pdfTeX-def|luatex-def>
1840 \MT@ifdefined@c@TF\MT@tr@ispace
1841 {\let\@tempa\MT@tr@ispace}%
1842 {\edef\@tempa{\MT@letterspace@*,,}}%
1843 \MT@ifdefined@c@TF\MT@tr@ospace
1844 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1845 {\edef\@tempa{\@tempa,,,}}%
1846 \expandafter\MT@tr@set@space\@tempa,%
1847 </pdfTeX-def|luatex-def>
1848 <*letterspace>
1849 % spacing = {<letterspace amount>*,,}
1850 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1851 * \fontdimen2\MT@lsfont/1000\relax
1852 </letterspace>
```

Adjust outer kerning (microtype only).

```
1853 <*pdfTeX-def|luatex-def>
1854 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,,}}%
1855 \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in letterspace).

```
1856 \MT@ifdefined@c@TF\MT@tr@ligatures\MT@tr@noligatures
1857 </pdfTeX-def|luatex-def>
1858 <*letterspace>
1859 % no ligatures = {f}
1860 \tagcode\MT@lsfont`f=\m@ne
1861 </letterspace>
```

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

```
1862 <luatex-def|letterspace> \MT@if@luaotf@font\relax{%
1863 <debug>\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1864 \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1865 \rprcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1866 \let\MT@the@pr@code\MT@the@pr@code@tr
1867 <luatex-def|letterspace> }%
1868 \fi
```

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

```
1869 \aftergroup\MT@set@lsfont
1870 <pdfTeX-def|luatex-def> \let\MT@font\MT@lsfont
1871 <luatex-def> \MT@if@luaotf@font\MT@font\relax
```

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

```
\MT@curr@ls 1872 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1873 \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.

```
1874 <*pdfTeX-def|luatex-def>
1875 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1876 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1877 \MT@tr@outer@l
1878 </pdfTeX-def|luatex-def>
```

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

```
1879 \ifx\MT@ls@adjust\empty
1880 <letterspace> % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1881 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1882 \MT@ls@outer@k
```

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

```
1883 <*pdfTeX-def|luatex-def>
1884 \else
1885 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1886 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1887 \ifdim\MT@outer@kern=\z@ \else \MT@ls@outer@k \fi
1888 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1889 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1890 </pdfTeX-def|luatex-def>
1891 <*letterspace>
1892 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1893 \MT@afteraftergroup{%
1894 \MT@set@curr@ok
1895 \noexpand\MT@ls@outer@k
1896 }%
1897 </letterspace>
1898 \fi
1899 <*pdfTeX-def|luatex-def>
```

`\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).

```
1900 \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.

```
1901 \MT@afteraftergroup{%
1902 \MT@set@curr@os
1903 \MT@set@curr@ok
1904 \noexpand\MT@tr@outer@r
1905 }%
1906 </pdfTeX-def|luatex-def>
1907 \fi
1908 <pdfTeX-def|luatex-def> }%
1909 }
```

`\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.)

```
1910 \def\MT@afteraftergroup#1{%
1911 <!letterspace> \MT@maybe@gobble@with@tikz{%
1912 \MT@ifdefined@n@TF{MT@aftergroup@number\currentgrouplevel}\relax%
```

```

1913     \MT@exp@cs\xdef{MT@aftergroup@number\currentgrouplevel}%
1914     {\MT@exp@cs\MT@gl@et{MT@aftergroup@number\currentgrouplevel}\noexpand\@undefined#1}%
1915     \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
1916     {MT@aftergroup@number\currentgrouplevel}%
1917     }%
1918     <!\letterspace> }%
1919 }
1920 </pdfTeX-def|luatex-def|letterspace>

\MT@ls@fontspec@colon    Add the kernfactor feature to a font loaded by fontspec (we might have to add
\MT@ls@fontspec@font    the colon ourselves).

1921 <*\luatex-def|letterspace>
1922 \def\MT@ls@fontspec@colon#1:#2:#3:#4@nil{\ifx\\#3\\#1:#2\else#1:#2:#3\fi}
1923 \def\MT@ls@fontspec@font#1 #2@nil{%
1924     "\MT@ls@fontspec@colon#1:::\relax\nil
1925     kernfactor=\MT@minus \ifnum\MT@letterspace@=1000 1\else 0.%
1926     \ifnum\MT@minus\MT@letterspace@<100 0\fi
1927     \ifnum\MT@minus\MT@letterspace@<10 0\fi
1928     \number\MT@minus\MT@letterspace@ \fi;"
1929     \ifx\\#2\\ at \f@size pt\else#2\fi\relax
1930 }
1931 </luatex-def|letterspace>

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

1932 <*\pdfTeX-def|luatex-def>
1933 \def\MT@get@tr@opt{%
1934     \MT@set@listname
1935     \let\MT@tr@factor@ \@m

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

1936 \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name @unit}{%
1937     \MT@let@cn\MT@tr@unit@{MT@tr@c@MT@tr@c@name @unit}%
1938     \ifdim\MT@tr@unit@=1em
1939         \let\MT@tr@unit@ \@undefined
1940     \else
1941         \MT@get@unit\MT@tr@unit@
1942         \fi
1943     }%
1944 \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name}{%
1945     \MT@let@cn\MT@letterspace{MT@tr@c@MT@tr@c@name}%
1946     \MT@ifdefined@c@T\MT@tr@unit@{%
1947         \let\@tempb\MT@letterspace
1948         \MT@scale@to@em
1949         \edef\MT@letterspace{\number\@tempcntb}%
1950     }%
1951     }%

\MT@tr@ispace    Adjust interword spacing.
\MT@tr@ospace 1952 \MT@get@tr@opt@{spacing} {ispace}%
1953 \MT@get@tr@opt@{outerspacing}{ospace}%

\MT@tr@okern    Adjust outer kerning.
1954 \MT@get@tr@opt@{outerkerning}{okern}%

\MT@tr@ligatures    Which ligatures should we disable (empty means all, undefined none)?
1955 \MT@get@tr@opt@{noligatures} {ligatures}%
1956 }

\MT@get@tr@opt@
1957 \def\MT@get@tr@opt@#1#2{%
1958     \MT@ifdefined@n@T{MT@tr@c@MT@tr@c@name @#1}%
1959     {\MT@let@nn{MT@tr@#2}{MT@tr@c@MT@tr@c@name @#1}}%
1960 }
1961 </pdfTeX-def|luatex-def>

```

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

```
1962 <pdfTeX-def|luatex-def|letterspace>
1963 <plain>\MT@requires@latex2{
1964 \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@currsz` 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@currsz`, fingers crossed. The overhead seems small.

```
1965 \DeclareRobustCommand\lsstyle{%
1966   \not@math@alphabet\lsstyle\textls
1967   \let\glb@currsz\@empty
1968   <pdfTeX-def|luatex-def> \MT@maybe@gobble@with@tikz{\aftergroup\glb@settings}%
1969   <pdfTeX-def|luatex-def> \def\MT@feat{tr}%
1970   \let\MT@tracking\MT@set@tr@codes
1971   \selectfont
1972 }
```

Now the definitions for the `letterspace` package with plain  $\text{T}_{\text{E}}\text{X}$ .

```
1973 <plain>
1974 }{
1975 \def\MT@set@lsfont{\MT@lsfont}
1976 \def\lsstyle{%
1977   \begingroup
1978   \escapechar\m@ne
1979   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1980   \MT@set@tr@codes
1981   \endgroup
1982 }
1983 \let\textls\@undefined
1984 \let\lslig\@undefined
1985 }
1986 </plain>
```

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

```
1987 \DeclareRobustCommand\lslig[1]{%
1988   {\MT@ifdefined@c@TF\MT@curr@ls{%
1989     \escapechar\m@ne
1990     \MT@get@ls@basefont
1991     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1992     \kern\MT@outer@kern
1993     \font@name #1%
1994     \kern\MT@outer@kern
1995   }}{#1}}%
1996 }
```

`\MT@ls@basefont`      pdf $\text{T}_{\text{E}}\text{X}$  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.

```
1997 \def\MT@get@ls@basefont{%
1998   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1999   \expandafter\ifx\MT@ls@basefont\relax
2000     \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
2001   \else
```

```

2002 <debug>\MT@info@n1{1}{... fixing base font}%
2003 \MT@exp@two@c\let\font@name\MT@ls@basefont
2004 \fi
2005 }

\MT@set@lsbasefont    If tracking is switched off in the middle of the document, or if \textls is called
\MT@set@tr@zero       with a zero letterspacing amount, we have to retrieve the base font and select it.
2006 \def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
2007 \def\MT@set@tr@zero{%
2008 <debug>\MT@info@n1{1}{... zero tracking}%
2009 \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
2010 \expandafter\ifx\MT@ls@basefont\relax \else
2011 <debug>\MT@info@n1{1}{... fixing base font}%
2012 \aftergroup\MT@set@lsbasefont
2013 \fi
2014 }
2015 </pdfTeX-def|luatex-def|letterspace>

\MT@tr@noligatures    pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.
2016 <*pdfTeX-def|luatex-def>
2017 <pdfTeX-def>\MT@requires@pdfTeX7{
2018 \def\MT@tr@noligatures{%
2019 \ifx\MT@tr@ligatures@empty
2020 \MT@noligatures@\MT@lsfont\@undefined
2021 \else
2022 \MT@noligatures@\MT@lsfont\MT@tr@ligatures
2023 \fi
2024 }
2025 <*pdfTeX-def>
2026 }{
2027 \def\MT@tr@noligatures{%
2028 \MT@warning@n1{%
2029 Disabling selected ligatures is only possible since\MessageBreak
2030 pdfTeX 1.40.4. Disabling all ligatures instead}%
2031 \MT@glet\MT@tr@noligatures\relax
2032 }
2033 }
2034 </pdfTeX-def>

\MT@outer@space       A new skip for outer spacing.
2035 \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.
2036 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
2037 <debug>\MT@info@n12{... orig. space: \the\fontdimen2\MT@lsfont,
2038 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
2039 <debug> \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
2040 \let\MT@temp@empty
2041 \MT@tr@set@space@{#1}{#4}{2}\@empty
2042 \MT@tr@set@space@{#2}{#5}{3}\@plus
2043 \MT@tr@set@space@{#3}{#6}{4}\@minus
2044 \MT@glet@c{MT@outer@space\expandafter\string\font@name}\MT@temp
2045 <debug>\MT@info@n12{... inner space: \the\fontdimen2\MT@lsfont,
2046 <debug> \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
2047 <debug>\MT@info@n12{... outer space: \MT@temp}%
2048 }

\MT@tr@set@space@     If settings for outer spacing (#2) don't exist, they will be inherited from the inner
                        spacing settings <#1>.
2049 \def\MT@tr@set@space@#1#2#3#4{%
2050 \MT@ifempty{#2}{%
2051 \MT@ifempty{#1}\relax{%

```



```

2052 \MT@tr@set@space@{#1}{#3}{1000}%
2053 \fontdimen#3\MT@lsfont=\@tempdima
2054 }%
2055 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
2056 }%
2057 \MT@tr@set@space@{#2}{#3}{2000}%
2058 \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2059 \MT@ifempty{#1}\relax{%
2060 \MT@tr@set@space@{#1}{#3}{1000}%
2061 \fontdimen#3\MT@lsfont=\@tempdima
2062 }%
2063 }%
2064 }

```

`\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.

```

2065 \def\MT@tr@set@space@{#1#2#3}%
2066 \MT@test@ast#1*\@nil{%
2067 \MT@ifdefined@c@TF\MT@tr@unit@
2068 {\edef\@tempb{#1}\MT@scale@to@em}
2069 {\@tempcntb=#1\relax}%
2070 \@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).

```

2071 \ifnum#2=\tw@
2072 \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
2073 \fi
2074 }%
2075 \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
2076 \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
2077 }%
2078 <debug>\MT@info{n13}{... : font dimen #2 (#1): \the\@tempdima}%
2079 }

```

`\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).

```

2080 \def\MT@tr@outer@{
2081 \ifhmode
2082 \ifdim\lastskip>5sp
2083 \edef\x{\the\lastskip minus 0pt}%
2084 \setbox\z@\hbox{\MT@outer@space=\x}%
2085 \ifdim\wd\z@>\z@
2086 <debug>\MT@info2{[[[ adjusting pre space: \the\MT@outer@space}%
2087 \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

2088 \let\MT@ls@outer@k\relax
2089 \else

```

The `ragged2e` package sets `\spaceskip` without glue.

```

2090 \ifdim\lastskip=
2091 \ifnum\spacefactor<2000
2092 \spaceskip
2093 \else
2094 \ifdim\xspaceskip=\z@
2095 \dimexpr\spaceskip+\fontdimen7\font@name\relax
2096 \else
2097 \xspaceskip
2098 \fi
2099 \fi
2100 <debug>\MT@info2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
2101 \unskip \hskip\MT@outer@space\relax
2102 \let\MT@ls@outer@k\relax

```

```

2103         \fi
2104     \fi
2105     \fi
2106     \fi
2107 }

```

`\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`).

```

2108 \def\MT@tr@outer@r{%
2109   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
2110 }

```

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

```

2111 \def\MT@if@outer@next#1{%
2112   \ifx\MT@tr@outer@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2113 }

```

`\MT@tr@outer@r@`

```

2114 \def\MT@tr@outer@r@{%
2115   \def\MT@temp*{}%

```

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

```

2116   \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).

```

2117     \ifnum\currentgrouptype=10 \else
2118       \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
2119 (debug)\MT@info2{}} adjusting post space (1): \the\MT@outer@space}%
2120       \fi}%
2121       \expandafter\ifcat\expandafter\noexpand\csname MT@tr@outer@next\endcsname\egroup
2122       \ifhmode\unkern\fi\egroup
2123       \MT@set@curr@ok \MT@set@curr@os
2124       \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}
2125     \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.

```

2126     \MT@if@outer@next\maybe@ic{%
2127       \MT@set@curr@ok \MT@set@curr@os
2128       \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}
2129     }{%

```

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).

```

2130     \MT@if@outer@next\check@icr{%
2131       \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}
2132     }{%
2133     \MT@if@outer@next\@sptoken{%
2134       \def\MT@temp* {\ifhmode\hskip\MT@outer@space
2135 (debug)\MT@info2{}} adjusting post space (2): \the\MT@outer@space}%
2136       \fi}%
2137     }{%
2138     \MT@if@outer@next~{%
2139       \def\MT@temp*~{\nobreak\hskip\MT@outer@space
2140 (debug)\MT@info2{}} adjusting post space (3): \the\MT@outer@space}%

```

```

2141             }%
2142         }{%
2143             \MT@if@outer@next\ \relax{%
2144             \MT@if@outer@next\space\relax{%
2145             \MT@if@outer@next\@xobeysp\relax{%

```

xspace requires special treatment.

```

2146             \MT@if@outer@next\xspace{%
2147             \def\MT@temp*\xspace{\MT@xspace}%
2148         }{%

```

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

```

2149             \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k
2150 (debug)\MT@dinfor2{--- adjusting post kern: \the\MT@outer@kern}%
2151             \fi}%
2152             \MT@let@nc{\MT@tr@outer@next}\relax
2153         }}}}}}\fi
2154     \fi\fi
2155     \MT@temp*%
2156 }

```

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

```

\MT@tr@outer@icr@ 2157 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
2158 \def\MT@tr@outer@icr@{%
2159     \let\@let@token= \MT@tr@outer@next
2160     \maybe@ic@
2161 }

```

\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.

```

2162 \def\MT@xspace{\futurelet\@let@token\MT@xspace@}
2163 \def\MT@xspace@{\@xspace@firsttrue\@xspace
2164     \ifdim\lastskip>5sp
2165         \unskip \hskip\MT@outer@space
2166     \else
2167         \ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k \fi
2168     \fi
2169 }

```

For older pdf<sub>T</sub>E<sub>X</sub> versions and Lua<sub>T</sub>E<sub>X</sub>, throw an error.

```

2170 }{
2171     \DeclareRobustCommand\lsstyle{%
2172         \MT@error{Letterspacing only works with \MT@engine tex version
2173 (pdfTex-def) 1.40%
2174 (luaTex-def) 0.62%
2175         \MessageBreak or newer}
2176         {Upgrade \MT@engine tex, or try the `soul' package instead.}%
2177         \MT@glet\lsstyle\relax
2178     }
2179 }

```

And for X<sub>Y</sub><sub>T</sub>E<sub>X</sub>, too.

```

2180 (/pdfTex-def|luaTex-def)
2181 (*xetTex-def)
2182 \DeclareRobustCommand\lsstyle{%
2183     \MT@error{Letterspacing currently doesn't work with xetex}
2184         {Run pdfTex or luaTex, or use the `soul' package instead.}%
2185     \MT@glet\lsstyle\relax
2186 }
2187 (/xetTex-def)

```

\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.

```

2188 <package|letterspace>
2189 \DeclareRobustCommand\textls{%
2190   \@ifstar{\let\MT@ls@adjust@empty\MT@textls}%
2191   {\let\MT@ls@adjust@empty\MT@ls@adjust@relax\MT@textls}%
2192 }

```

`\MT@textls` This is now almost L<sup>A</sup>T<sub>E</sub>X's `\DeclareTextFontCommand`, with the difference that we  
`\MT@letterspace@` adjust the outer spacing and kerning also for `\lsstyle`, while L<sup>A</sup>T<sub>E</sub>X's text switches  
 don't bother about italic correction.

```

2193 \newcommand\MT@textls[2][{}]{%
2194   \ifmmode
2195     \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
2196   \else
2197     \hmode@bgroup
2198       \MT@ls@set@ls{#1}%
2199       \lsstyle #2%
2200     \expandafter
2201     \egroup
2202   \fi
2203 }

```

`\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 2204 \def\MT@ls@adjust@empty{\let\MT@ls@adjust@empty}
\MT@ls@set@ls       2205 \def\MT@ls@adjust@relax{\let\MT@ls@adjust@relax}
2206 \def\MT@ls@set@ls#1{%
2207   \MT@ifempty{#1}%
2208   {\let\MT@letterspace@{\undefined}%
2209   {\KV@spdef\MT@letterspace@{#1}%
2210   \edef\MT@letterspace@{\number\MT@letterspace@}%
2211   \MT@ls@too@large\MT@letterspace@}%
2212   \MT@ls@adjust@
2213 }

```

`\MT@ls@too@large` Test whether letterspacing amount is too large.

```

2214 \def\MT@ls@too@large#1{%
2215   \ifnum#1>\MT@tr@max
2216     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
2217     \let#1\MT@tr@max
2218   \else
2219     \ifnum#1<\MT@tr@min
2220       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
2221       \let#1\MT@tr@min
2222     \fi
2223   \fi
2224 }

```

`\MT@outer@kern` This dimen is used for the starred version of `\textls`, for `\lslig` and for adjusted  
`\MT@tr@set@okern` outer kerning.

```

2225 \newdimen\MT@outer@kern
2226 </package|letterspace>
2227 <pdfTeX-def|luatex-def>
2228 \def\MT@tr@set@okern#1,#2,{%
2229   \let\MT@temp@empty
2230   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
2231   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
2232   \MT@gl@et@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
2233   <debug>\MT@edinfo{n12{... outer kerning: (#1,#2)
2234   <debug>           = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%
2235 }

```

`\MT@tr@set@okern@`

```

2236 \def\MT@tr@set@okern@#1{%
2237   \MT@test@ast#1*\@nil{%
2238     \MT@ifdefined@c@TF\MT@tr@unit@
2239     {\edef\@tempb{#1}\MT@scale@to@em}
2240     {\@tempcntb=#1\relax}%
2241     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
2242   }{%
2243     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
2244     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
2245     * \fontdimen6\MT@lsfont/2000\relax
2246   }%
2247   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
2248   * \fontdimen6\MT@lsfont/2000\relax
2249   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
2250 }
2251 </pdfTeX-def|luatex-def>

```

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

```

2252 <(*pdfTeX-def|luatex-def|letterspace)>
2253 \def\MT@ls@outer@k{%
2254   \ifhmode
2255     \ifdim\lastkern=-3sp \unkern
2256     \ifdim\lastkern=3sp \kern-3sp
2257       \expandafter\expandafter\expandafter\@gobble
2258     \else \unkern
2259       \expandafter\expandafter\expandafter\@firstofone
2260     \fi
2261   \else
2262     \expandafter\@firstofone
2263   \fi
2264   {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
2265   \fi
2266 }
2267 </pdfTeX-def|luatex-def|letterspace>

```

### 14.2.6 Disabling ligatures

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

```

2268 <(*pdfTeX-def|luatex-def)>
2269 <pdfTeX-def>\MT@requires@pdfTeX5{
2270 \def\MT@noligatures{%
2271   \MT@dotrue
2272   \let\@tempa\MT@nl@setname
2273   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
2274     \MT@ifdefined@c@TF{\MT@checklist@##1}%
2275     {\csname MT@checklist@##1\endcsname}%
2276     {\MT@checklist@{##1}}}%
2277   {nl}%
2278   }%
2279   \ifMT@do
2280     \MT@noligatures@\MT@font\MT@nl@ligatures
2281   \fi
2282 }

```

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

```

2283 <luatex-def>\MT@requires@luatex4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
2284 \def\MT@noligatures@#1#2{%
2285   \MT@ifdefined@c@TF#2%

```

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

```

2286   \MT@ifdefined@c@TF\tagcode%

```

No ‘inputenc’ key.

```

2287 \let\MT@warn@maybe@inputenc\empty
2288 \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
2289 \MT@map@clist@c#2{%
2290 \KV@sp@def\@tempa{##1}\MT@get@slot
2291 \ifnum\MT@char>\m@ne
2292 \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`).

```

2293 <luatex-def> \MT@if@luaotf@font
2294 <luatex-def> {\MT@lua{microtype.noligatures([[#1]],[[\MT@char]])}}\relax
2295 \fi
2296 }%
2297 \MT@vinfo{... Disabling ligatures for characters: #2}%
2298 }{%
2299 \pdfnoligatures#1%
2300 \MT@warning{Cannot disable selected ligatures (pdfTeX doesn't\MessageBreak
2301 know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
2302 the font instead}%
2303 }%
2304 }{%
2305 \pdfnoligatures#1%
2306 <luatex-def> \MT@if@luaotf@font
2307 <luatex-def> {\MT@lua{microtype.noligatures([[#1]],"_all_")}}\relax
2308 \MT@vinfo{... Disabling all ligatures}%
2309 }%
2310 }
2311 <pdfTeX-def>\relax
2312 </pdfTeX-def|luatex-def>

```

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@`.

```

2313 <*luafile>
2314 microtype.ligs = microtype.ligs or { }
2315
2316 local function noligatures(fontcs,liga)
2317 local fontcs = match(fontcs,"([^\s]+)")
2318 microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
2319 table.insert(microtype.ligs[fontcs],liga)
2320 end
2321 microtype.noligatures = noligatures
2322
2323 local function keepligature(c)
2324 local nodedirect = node.direct
2325 local getfield = nodedirect.getfield
2326 local getfont = nodedirect.getfont
2327 local f,ch
2328 if type(c) == "userdata" then -- in older luaotfload versions, c was a node
2329 f = c.font
2330 ch = c.components.char
2331 else -- since 2.6, c is a (direct node) number
2332 f = getfont(c)
2333 ch = getfield(getfield(c,"components"),"char")
2334 end
2335 -- if ch then -- should always be true
2336 local ligs = microtype.ligs[match(tex.fontidentifier(f),"\\([^\s]+)")]
2337 if ligs then
2338 for _,lig in pairs(ligs) do
2339 if lig == "_all_" or tonumber(lig) == ch then
2340 return false
2341 end

```

```

2342     end
2343   end
2344   return true
2345 -- end
2346 end
2347
2348 if luaotfload and luaotfload.letterspace then
2349   if luaotfload.letterspace.keepligature then
2350     microtype.info("overwriting function `keepligature'")
2351   end
2352   luaotfload.letterspace.keepligature = keepligature
2353 end
2354
2355 /luafile

```

### 14.2.7 Loading the configuration

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

```

2356 <*package>
2357 \def\MT@load@list#1{%
2358   \edef\@tempa{#1}%
2359   \MT@let@cn\@tempb{\MT@MT@feat @c\@tempa @load}%
2360   \MT@ifstreq\@tempa\@tempb{%
2361     \MT@error{\@nameuse{\MT@abbr\MT@feat} list `'\@tempa' cannot load itself}{}%
2362   }%
2363   \ifx\@tempb\relax \else
2364     \MT@ifdefinedn@TF{\MT@MT@feat @c\@tempb}{%
2365       \MT@vinfo{... : First loading \@nameuse{\MT@abbr\MT@feat} list `'\@tempb'}%
2366       \begingroup
2367         \MT@load@list\@tempb
2368       \endgroup
2369       \edef\MT@curr@list@name{\@nameuse{\MT@abbr\MT@feat} list
2370         \noexpand\MessageBreak`'\@tempb'}%
2371       \MT@let@cn\@tempc{\MT@MT@feat @c\@tempb}%
2372       \expandafter\MT@set@codes\@tempc,\relax,%
2373     }{%
2374       \MT@error{\@nameuse{\MT@abbr\MT@feat} list `'\@tempb' undefined.\MessageBreak
2375         Cannot load it from list `'\@tempa'}{}%
2376     }%
2377   \fi
2378 }%
2379 }

```

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

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

```

2380 \let\MT@file@list\empty
2381 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

2382   \MT@in@clist{#1}\MT@file@list
2383   \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.

```

2384   \MT@begin@catcodes
2385   \let\MT@begin@catcodes\relax
2386   \let\MT@end@catcodes\relax
2387   \InputIfFileExists{mt-#1.cfg}{%
2388     \edef\MT@curr@file{mt-#1.cfg}%
2389     \MT@vinfo{... Loading configuration file \MT@curr@file}%
2390     \MT@xadd\MT@file@list{#1,%
2391   }{%
2392     \MT@get@basefamily#1\empty\empty\empty\empty

```

```

2393      \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2394      \ifMT@inlist@
2395      \MT@xadd\MT@file@list{#1,}%
2396      \else
2397      \InputIfFileExists{mt-\@tempa.cfg}{%
2398      \edef\MT@curr@file{mt-\@tempa.cfg}%
2399      \MT@vinfo{... Loading configuration file \MT@curr@file}%
2400      \MT@xadd\MT@file@list{\@tempa,#1,}%
2401      }{%
2402      \MT@vinfo{... No configuration file mt-#1.cfg}%
2403      \MT@xadd\MT@file@list{#1,}%
2404      }%
2405      \fi
2406      }%
2407      \endgroup
2408      \fi
2409      }

```

`\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 14.1.5.)

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

```

2410 \def\MT@cfg@catcodes{%
2411   \makeatletter
2412   \catcode\^7%
2413   \catcode\ 9%
2414   \catcode\^^I9%
2415   \catcode\^^M9%
2416   \catcode\\\z@
2417   \catcode\{\@ne
2418   \catcode\}\tw@
2419   \catcode\%6%
2420   \catcode\%14%
2421   \MT@map@tlist@n
2422   {\!"\$&'\"()*+,-./:;<=>?[\]\_`{|}~}%
2423   \@makeother
2424 }

```

`\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.

```

2425 \def\MT@begin@catcodes{%
2426   \begingroup
2427   \MT@cfg@catcodes
2428 }

```

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

```

2429 \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.

```

2430 \def\MT@get@basefamily#1#2#3#4\nil{%
2431   \ifx\@empty#4%
2432   \def\@tempa{#1#2#3}%
2433   \else
2434   \let\@tempa\@empty

```



Table 4:

Order for matching font attributes

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Encoding	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Family	•	•	•	•	•	•	•	•	-	-	-	-	-	-	-	-
Series	•	•	•	•	-	-	-	-	•	•	•	•	-	-	-	-
Shape	•	•	-	-	•	•	-	-	•	•	-	-	•	•	-	-
Size	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```

2435 \edef\@tempb{#1#2#3#4}%
2436 \expandafter\MT@get@basefamily@\@tempb\@nil
2437 \fi
2438 }

```

\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’.

```

2439 \def\MT@get@basefamily@#1#2\@nil{%
2440 \edef\@tempa{\@tempa#1}%
2441 \ifx\#2\expandafter\gobble\else\expandafter\@firstofone\fi
2442 {\MT@in@tlist{#2}\MT@variants
2443 \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
2444 }

```

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

```

\MT@get@listname
\MT@get@listname@
\MT@get@listname@
2445 \def\MT@get@listname#1{%
2446 (debug)\MT@info{n}{1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
2447 \let\MT@listname\@undefined
2448 \def\@tempb{#1}%
2449 \MT@map@tlist@c\MT@try@order\MT@get@listname@
2450 }
2451 \def\MT@get@listname@#1{%
2452 \expandafter\MT@next@listname#1%
2453 \ifx\MT@listname\@undefined \else
2454 \expandafter\MT@tlist@break
2455 \fi
2456 }

```

\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 4 in the documentation part any longer and can cast it off here.

```

2457 \def\MT@try@order{%
2458 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2459 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2460 }

```

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

```

2461 \def\MT@next@listname#1#2#3#4{%
2462 \ifnum#1=\z@\MT@nofamilytrue\fi
2463 \edef\@tempa{\MT@encoding
2464 /\ifnum#1=\@ne \MT@family \fi
2465 /\ifnum#2=\@ne \MT@series \fi
2466 /\ifnum#3=\@ne \MT@shape \fi
2467 /\ifnum#4=\@ne *\fi
2468 \MT@context}%
2469 (debug)\MT@info{n}{1}{trying \@tempa}%
2470 \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2471 \MT@next@listname@#4%
2472 }%

```

Also try with an alias family.

```

2473 \ifnum#1=\@ne
2474 \ifx\MT@familyalias\@empty \else
2475 \edef\@tempa{\MT@encoding
2476 \MT@familyalias
2477 /\ifnum#2=\@ne \MT@series\fi
2478 /\ifnum#3=\@ne \MT@shape\fi
2479 /\ifnum#4=\@ne *\fi
2480 \MT@context}%
2481 (debug)\MT@info@n1{1}{(alias) \@tempa}%
2482 \MT@ifdefined@n@T{\MT@\@tempb @\@tempa}{%
2483 \MT@next@listname@#4%
2484 }%
2485 \fi
2486 \fi
2487 }%
2488 }

```

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

```

2489 \def\MT@next@listname@#1{%
2490 \ifnum#1=\@ne
2491 \MT@exp@cs\MT@in@list{\MT@\@tempb @\@tempa @sizes}%
2492 \ifMT@in@list@
2493 \let\MT@listname\MT@size@name
2494 \fi
2495 \else
2496 \MT@let@cn\MT@listname{\MT@\@tempb @\@tempa}%
2497 \fi
2498 }

```

\MT@if@list@exists

```

\MT@context 2499 \def\MT@if@list@exists{%
2500 \MT@let@cn\MT@context{\MT@\MT@feat @context}%
2501 \MT@ifstreq{0}\MT@context{\let\MT@context\@empty}\relax
2502 \MT@get@listname{\MT@feat @c}%
2503 \MT@ifdefined@c@TF\MT@listname{%
2504 \MT@edef@n{\MT@\MT@feat @c@name}{\MT@listname}%
2505 \ifMT@nonselected
2506 \MT@vinfo{... Applying non-selected expansion (list `'\MT@listname')}%
2507 \else
2508 \MT@vinfo{... Loading \@nameuse{\MT@abbr@\MT@feat} list `'\MT@listname'}%
2509 \fi
2510 \@firstoftwo
2511 }{%

```

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

```

2512 \MT@let@cn{\MT@\MT@feat @c@name}\@empty

```

Don't warn if selected=false.

```

2513 \ifMT@nonselected
2514 \MT@vinfo{... Applying non-selected expansion (no list)}%
2515 \else

```

Tracking doesn't require a list, either.

```

2516 \MT@ifstreq\MT@feat{tr}\relax{%
2517 \MT@warning{I cannot find a \@nameuse{\MT@abbr@\MT@feat} list
2518 for font\MessageBreak'\MT@font'%
2519 \ifx\MT@context\@empty\else\space(context: `'\MT@context')}\fi.
2520 Switching off\MessageBreak\@nameuse{\MT@abbr@\MT@feat} for this font}%
2521 }%
2522 \fi
2523 \@secondoftwo
2524 }%

```

```

2525 }
\MT@get@inh@list    The inheritance lists are global (no context).
\MT@context 2526 \def\MT@get@inh@list{%
2527   \let\MT@context\@empty
2528   \MT@get@listname{\MT@feat @inh}%
2529   \MT@ifdefined@c@TF\MT@listname{%
2530     \MT@edef\MT@MT@feat @inh@name{\MT@listname}%
2531     (debug) \MT@info{1}{... Using \@nameuse{\MT@abbr@\MT@feat} inheritance list
2532     (debug) \MT@listname}%
2533     \MT@let\MT@tempc{\MT@MT@feat @inh@\MT@listname}%
    If the list is \@empty, it has already been parsed.
2534     \ifx\MT@tempc\@empty \else
2535     (debug) \MT@info{1}{parsing inheritance list ...}%
    The group is only required in case an input encoding is given.
2536     \begingroup
2537     \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
2538     \MT@set@inputenc{inh}%
2539     \expandafter\MT@inh@do\MT@tempc,\relax,%
2540     \MT@gl@et@nc{\MT@MT@feat @inh@\MT@listname}\@empty
2541     \endgroup
2542     \fi
2543   }%
2544   \MT@let\MT@MT@feat @inh@name\@undefined
2545 }%
2546 }

```

#### 14.2.8 Translating characters into slots

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

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

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

```

\MT@char@ 2547 \def\MT@get@slot{%
2548   \escapechar~\
2549   \let\MT@char@mone
2550   \MT@noresttrue

```

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

```

2551 \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.

```

2552 \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.

```

2553 \expandafter\MT@is@letter\@tempa\relax\relax
2554 \ifnum\MT@char@ < \z@

```

- OK, so it must be a macro. We do not allow random commands but only those defined in L<sup>A</sup>T<sub>E</sub>X'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.

```
2555 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2556 \MT@is@symbol
```

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

```
2557 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2558 \ifnum\MT@char@ < \z@
```

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

```
2559 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2560 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2561 \fi
2562 \fi

2563 \let\MT@char\MT@char@
2564 \MT@get@slot@
2565 \escapechar\m@ne
2566 }
2567 </package>
```

`\MT@get@slot@`

```
2568 <pdfTeX-def|luatex-def|xetex-def>
2569 \def\MT@get@slot@{%
```

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

```
2570 <xetex-def> \ifnum\XeTeXfonttype\MT@font=\z@
2571 \ifnum\MT@char > \m@ne
```

In Lua<sub>T</sub><sub>E</sub><sub>X</sub>, it may also be a glyph name, prefixed with `'`.

```
2572 <luatex-def>
2573 \ifnum\MT@char=47\relax
2574 \ifMT@noest \else
2575 \@tempcnta=\MT@lua{
2576   local glyph = microtype.name_to_slot([[ \expandafter\@gobble\@tempa ]],true)
2577   if glyph then tex.write(glyph)
2578   else tex.write(-1)
2579   end
2580 } \relax
2581 \ifnum\@tempcnta<\z@
2582 \MT@warn@unknown
2583 \let\MT@char\m@ne
2584 \else
2585 \edef\MT@char{\the\@tempcnta}%
2586 <debug>\MT@info@n1{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
2587 \fi
2588 \fi
2589 \else
2590 </luatex-def>
```

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.

```
2591 \ifMT@noest \else
2592 \MT@warn@rest
2593 <pdfTeX-def|luatex-def> \let\MT@char\m@ne
2594 <xetex-def> \let\MT@char\@empty
2595 \fi
2596 <luatex-def> \fi
2597 \else
2598 \MT@warn@unknown
```

```

2599 <xetex-def> \let\MT@char\empty
2600 \fi
2601 <*xetex-def>
2602 \else

```

There are more possibilities for X<sub>Y</sub>TeX: It may also be a glyph name (prefixed with ‘/’). We indicate this to \MT@get@charwd by reversing the sign of \MT@char@.

```

2603 \ifnum\MT@char=47\relax
2604 \ifMT@norest \edef\MT@char{U47}%
2605 \else
2606 \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
2607 \ifnum\@tempcnta=\z@
2608 \MT@warn@unknown
2609 \let\MT@char\empty
2610 \else
2611 \edef\MT@char{\@tempa\space}%
2612 \edef\MT@char@{-\the\@tempcnta}%
2613 <debug>\MT@info@n1{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
2614 \fi
2615 \fi
2616 \else
2617 \ifnum\MT@char > \m@ne
2618 \ifMT@norest

```

Or, it’s a Unicode number, which we mustn’t translate into a glyph number, since the latter is font-specific.

```

2619 \@tempcnta=\XeTeXcharglyph\MT@char\relax
2620 \ifnum\@tempcnta=\z@
2621 \MT@info@missing@char
2622 \let\MT@char\empty
2623 \else
2624 <debug>\MT@info@n1{3}{> (glyph number: \the\@tempcnta,
2625 <debug> glyph name: \XeTeXglyphname\MT@font\@tempcnta)}%
2626 \edef\MT@char{U\MT@char}%
2627 \fi
2628 \else
2629 \MT@warn@rest
2630 \let\MT@char\empty
2631 \fi
2632 \else
2633 \MT@warn@unknown
2634 \let\MT@char\empty
2635 \fi
2636 \fi
2637 \fi
2638 </xetex-def>
2639 }
2640 </pdfTEX-def|luaTEX-def|xetex-def>

```

This is the lua function to translate glyph name into slot number. Beginning with v2.2, luaotfload provides this function in an API, which we use if available, but (for now, at least) keep the old code for backward compatibility.

```

2641 <*luafile>
2642 if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
2643 local slot_of_name = luaotfload.aux.slot_of_name
2644 microtype.name_to_slot = function(name, unsafe)
2645 return slot_of_name(font.current(), name, unsafe)
2646 end
2647 else
2648 -- we dig into internal structure (should be avoided)
2649 local function name_to_slot(name, unsafe)
2650 if fonts then
2651 local unicodes
2652 if fonts.ids then -- legacy luaotfload

```

```

2653     local tfmdata = fonts.ids[font.current()]
2654     if not tfmdata then return end
2655     unicodes = tfmdata.shared.otfdata.luatex.unicodes
2656   else -- new location
2657     local tfmdata = fonts.hashes.identifiers[font.current()]
2658     if not tfmdata then return end
2659     unicodes = tfmdata.resources.unicodes
2660   end
2661   local unicode = unicodes[name]
2662   if unicode then -- does the 'or' branch actually exist?
2663     return type(unicode) == "number" and unicode or unicode[1]
2664   end
2665 end
2666 end
2667 microtype.name_to_slot = name_to_slot
2668 end
2669
2670 /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** 2671 *<pdfTeX-def|luatex-def|xetex-def>*  
2672 *\def\MT@max@char*  
2673 *<pdfTeX-def> {127 }*  
2674 *<luatex-def|xetex-def> {1114111 }*  
2675 *\def\MT@max@slot*  
2676 *<pdfTeX-def> {255 }*  
2677 *<luatex-def|xetex-def> {1114111 }*  
2678 *<pdfTeX-def|luatex-def|xetex-def>*

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

```

2679 <package>
2680 \newif\ifMT@noreset

2681 \def\MT@is@letter#1#2\relax{%
2682   \ifcat a\noexpand#1\relax
2683     \edef\MT@char@\number`#1}%
2684     \ifx\#2\%
2685 <debug>\MT@info@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2686     \else
2687       \MT@noresetfalse
2688     \fi
2689   \else
2690     \ifcat !\noexpand#1\relax
2691       \edef\MT@char@\number`#1}%
2692 <debug>\MT@info@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
2693       \ifx\#2\%
2694         \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
2695       \else
2696         \MT@noresetfalse
2697         \expandafter\MT@is@number#1#2\relax\relax
2698       \fi
2699     \fi
2700   \fi
2701 }

```

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

```

2702 \def\MT@is@number#1#2#3\relax{%
2703   \ifx\relax#3\relax \else
2704     \ifx\relax#2\relax \else
2705       \MT@noresettrue

```

```

2706     \if#1"\relax
2707     \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2708 (debug)\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2709     \else
2710     \if#1'\relax
2711     \def\MT@char@{\number#1#2#3}%
2712 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2713     \else
2714     \MT@ifint{#1#2#3}{%
2715     \def\MT@char@{\number#1#2#3}%
2716 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2717     }\MT@noestfalse
2718     \fi
2719     \fi
2720     \ifnum\MT@char@ > \MT@max@slot
2721     \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2722     \let\MT@char@\m@ne
2723     \fi
2724     \fi
2725     \fi
2726 }

```

`\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., Å 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.

```

2727 \def\MT@is@active#1#2\@nil{%
2728   \ifnum\catcode`#1 = \active
2729     \begingroup
2730     \set@display@protect
2731     \let\IeC\@firstofone
2732     \let\@inpc@undefined@\MT@undefined@char

```

Unicode handling has changed again with L<sup>A</sup>T<sub>E</sub>X 2019/10/01.

```

2733   \let\UTF@two@octets@noexpand\@empty
2734   \let\UTF@three@octets@noexpand\@empty
2735   \let\UTF@four@octets@noexpand\@empty

```

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

```

2736   \def\UTFviii@defined##1{\ifx ##1\relax
2737     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%

```

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

```

2738   \MT@ifdefined@c@T\PrerenderUnicode
2739     {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%

```

The `\expandafter` hocus-pocus should please `newunicodechar`.

```

2740   \edef\x{\endgroup
2741     \def\noexpand\@tempa{\expandafter\expandafter\expandafter\@empty\@tempa}%

```

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

```

2742   \MT@toks={\the\MT@toks\space=
2743     \expandafter\expandafter\expandafter\@empty\@tempa}%
2744   }%
2745   \x
2746   \fi
2747 }

```

<code>\MT@undefined@char</code>	For characters not defined in the current input encoding.
2748	<code>\def\MT@undefined@char#1{undefined in input encoding ``#1''}</code>
<code>\MT@is@symbol</code>	The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding <code>\&lt;command&gt;</code> , we construct the command <code>\&lt;encoding&gt;\&lt;command&gt;</code> and see whether its meaning is <code>\char"&lt;hex number&gt;</code> , which is the case for everything that has been defined with <code>\DeclareTextSymbol</code> in the encoding definition files.
2749	<code>\def\MT@is@symbol{%</code>
2750	<code>\expandafter\def\expandafter\MT@char\expandafter</code>
2751	<code>{\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%</code>
2752	<code>\expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter</code>
2753	<code>\meaning\expandafter\MT@char\MT@charstring\relax\relax\relax</code>
2754	<code>\ifnum\MT@char@ &lt; \z@</code>
	For TU encoding, the commands <code>\textquotesingle</code> , <code>\textasciigrave</code> and <code>\textquotedbl</code> are defined by means of the auxiliary macro <code>\remove@tlig</code> , which we take care of here.
2755	<code>\expandafter\expandafter\expandafter\MT@is@tlig\MT@char\relax\relax</code>
2756	<code>\ifnum\MT@char@ &lt; \z@</code>
	Finally, if it hasn't been defined by <code>\DeclareTextSymbol</code> , it could be a letter (e.g., <code>\i</code> , when using frenchpro).
2757	<code>\expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax</code>
2758	<code>\fi</code>
2759	<code>\fi</code>
2760	<code>}</code>
<code>\MT@is@char</code>	A helper macro that inspects the <code>\meaning</code> of its argument.
<code>\MT@charstring</code>	2761 <code>\begingroup</code>
	2762 <code>\catcode`\=/\z@</code>
	2763 <code>/\MT@map@tlist@n{\CHARLEX}/@makeother</code>
	2764 <code>/lowercase{%</code>
	2765 <code>/def/x{/endgroup</code>
	2766 <code>/def\MT@charstring{\CHAR"%</code>
	2767 <code>/def\MT@is@char##1\CHAR"##2##3##4/relax{%</code>
	2768 <code>/ifx/relax##4/relax</code>
	2769 <code>/ifMT@xunicode</code>
	2770 <code>/expandafter\MT@is@charx\MT@strip@prefix##1&gt;/relax\CHAR "%</code>
	2771 <code>/relax/relax/relax/relax/relax</code>
	2772 <code>/fi</code>
	2773 <code>/else</code>
	2774 <code>/ifx/relax##1/relax</code>
	2775 <code>/if##3\relax</code>
	2776 <code>/edef\MT@char@{/number"##2}%</code>
	2777 <code>/MT@ifstreq\MT@charstring{##3##4}/relax\MT@noestfalse</code>
	2778 <code>/else</code>
	2779 <code>/edef\MT@char@{/number"##2##3}%</code>
	2780 <code>/MT@ifstreq\MT@charstring{##4}/relax</code>
	2781 <code>{/MT@is@xchar##2##3 ##4\CHAR"/relax}%</code>
	2782 <code>/fi</code>
	2783 <code>&lt;debug&gt; /MT@dinfo@n1{3}{&gt; ~/the\MT@toks' is a \char (/MT@char@)}%</code>
	2784 <code>/fi</code>
	2785 <code>/fi</code>
	2786 <code>}%</code>
<code>\MT@is@xchar</code>	With fontspec's TU encoding, glyph numbers may be up to four digits.
2787	<code>/def\MT@is@xchar##1 ##2\CHAR"##3##4/relax{%</code>
2788	<code>/MT@ifstreq\MT@charstring{##3##4}%</code>
2789	<code>{/edef\MT@char@{/number"##1##2}/MT@noestfalse</code>
2790	<code>}%</code>
<code>\MT@charxstring</code>	For xunicode, which doesn't <code>\countdef</code> , but rather <code>\def</code> s the chars.
<code>\MT@strip@prefix</code>	
<code>\MT@is@charx</code>	



```

2791 /def/MT@charxstring{\CHAR "%}
2792 /def/MT@strip@prefix##1>##2/relax{##2}%
2793 /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
2794 /ifx/relax##1/relax
2795 /ifx/relax##6/relax/else
2796 /edef/MT@char@{/number"##2##3##4##5}%
2797 /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@noestfalse
2798 (debug) /MT@dinfo@n1{3}{> `the/MT@toks' is a xunicode \char (/MT@char@)}%
2799 /fi
2800 /fi
2801 }%
2802 }%
2803 }
2804 /x

```

`\MT@is@tlig` This might have to change again with the next L<sup>A</sup>T<sub>E</sub>X release, ... or so I feared, but it still seems to be fine.

```

2805 \def\MT@is@tlig#1#2{%
2806 \ifx#1\remove@tlig
2807 (debug) \MT@dinfo@n1{3}{> `the/MT@toks' (removing remove@tlig)}%
2808 \MT@is@number #2\relax\relax
2809 \fi
2810 }

```

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

```

2811 \def\MT@is@composite#1#2\relax{%
2812 \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`.

```

2813 \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2814 \string\csname\MT@encoding\endcsname
2815 \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%

```

In 2017, L<sup>A</sup>T<sub>E</sub>X introduced a new way of declaring accented Unicode commands (`\DeclareUnicodeComposite`), which we take care of here (`\UnicodeEncodingName` has been introduced at the same time):

```

2816 \ifx\UnicodeEncodingName\undefined\else
2817 \expandafter\expandafter\expandafter
2818 \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
2819 \fi
2820 \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax

```

Again, xunicode.

```

2821 \ifnum\MT@char@ < \z@
2822 \ifMT@xunicode
2823 \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
2824 \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
2825 \MT@char\MT@charxstring\relax\relax\relax\relax\relax
2826 \fi
2827 \fi
2828 \fi
2829 }

```

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

```

2830 \def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
2831 \ifx\#2\\\else\edef\MT@char{\iffontchar#2\fi}\fi
2832 }

```

[What about math? Well, for a moment the following looked like a solution, with

`\mt@is@mathchar` defined accordingly, analogous to `\MT@is@char` above, to pick up the last two tokens (the `\meaning` of a `\mathchardef`'ed command expands to its hexadecimal notation):

```
\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode~#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
    \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}
```

However, the problem is that `\mathcodes` and `\mathchardefs` have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e.g., the minus in `cmsy` when the `euler` package is loaded). So, no way to go, unfortunately.]

Some warning messages, for performance reasons separated here.

`\MT@curr@list@name` The type and name of the current list, defined at various places.

```
\MT@set@listname 2833 \def\MT@set@listname{%
2834   \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2835     \@nameuse{MT@\MT@feat @c@name}}}%
2836 }
```

`\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.

```
2837 \def\MT@warn@ascii{%
2838   \MT@warning@nl{Character `the\MT@toks' (= \MT@char@)
2839     is outside of ASCII range.\MessageBreak
2840     You must load the `inputenc' package before using\MessageBreak
2841     8-bit characters in \MT@curr@list@name}%
2842 }
```

`\MT@warn@number@too@large` Number too large.

```
2843 \def\MT@warn@number@too@large#1{%
2844   \MT@warning@nl{%
2845     Number #1 in encoding `MT@encoding' too large!\MessageBreak
2846     Ignoring it in \MT@curr@list@name}%
2847 }
```

`\MT@warn@rest` Not all of the string has been parsed.

```
2848 \def\MT@warn@rest{%
2849   \MT@warning@nl{%
2850     Unknown slot number of character\MessageBreak`the\MT@toks'%
2851     \MT@warn@maybe@inputenc\MessageBreak
2852     in font encoding `MT@encoding'.\MessageBreak
2853     Make sure it's a single character\MessageBreak
2854     (or a number) in \MT@curr@list@name}%
2855 }
```

`\MT@warn@unknown` No idea what went wrong.

```
2856 \def\MT@warn@unknown{%
2857   \MT@warning@nl{%
2858     Unknown slot number of character\MessageBreak`the\MT@toks'%
2859     \MT@warn@maybe@inputenc\MessageBreak
2860     in font encoding `MT@encoding' in \MT@curr@list@name}%
2861 }
```

`\MT@warn@maybe@inputenc` In case an input encoding had been requested.

```
2862 \def\MT@warn@maybe@inputenc{%
2863   \MT@ifdefined@n@T
```

```

2864 {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2865 { (input encoding ~\@nameuse
2866 {MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2867 }

```

#### 14.2.9 Hook into L<sup>A</sup>T<sub>E</sub>X's font selection

We append `\MT@setupfont` to `\pickup@font`, which is called by L<sup>A</sup>T<sub>E</sub>X every time a font is selected. We then check whether we've already seen this font, and if not, set it up for micro-typography. This ensures that we will catch all fonts, and that we will not set up fonts more than once. The whole package really hangs on this command.

In contrast to the `pdfcprot` package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up.

For my reference:

- `\pickup@font` is called by `\selectfont`, `\wrong@fontshape`, or `\getanddefine@fonts` (for math).
- `\pickup@font` calls `\define@newfont`.
- `\define@newfont` may call (inside a group!)
  - `\wrong@fontshape`, which in turn will call `\pickup@font`, and thus `\define@newfont` again, or
  - `\extract@font`.
- `\get@external@font` is called by `\extract@font`, by itself, and by the substitution macros.

Up to version 1.3 of this package, we were using `\define@newfont` as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before `microtype` and were loading fonts, e.g., `jurabib`, `ledmac`, `pifont` (loaded by `hyperref`), `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 2868 \let\MT@font@list\empty
2869 \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`.

```

2870 </package>
2871 <*package|letterspace>
2872 <plain>\MT@requires@latex2{
2873 \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.

```
2874 (package) \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
2875 (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.

```
2876 \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.

```
2877 \ifpackage@loaded{xeCJK}{\@firstofone}{%
2878 \ifpackage@later{CJK}{2006/10/17}% 4.7.0
2879 {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
2880 {\def\MT@orig@pickupfont{\ifundefined{CJK@plane}}}%
2881 \g@addto@macro\MT@orig@pickupfont
2882 {\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).

```
2883 \ifpackage@loaded{CJKutf8}%
2884 {\ifpackage@later{CJKutf8}{2008/05/22}% 4.8.0
2885 {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
2886 {\@firstoftwo}}%
2887 {\@firstoftwo}%
2888 {\g@addto@macro\MT@orig@pickupfont{%
2889 {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2890 \define@newfont\else\xdef\font@name{%
2891 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2892 {\g@addto@macro\MT@orig@pickupfont{%
2893 {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2894 \define@newfont\def\CJK@temp{v}%
2895 \ifx\CJK@temp\CJK@plane
2896 \expandafter\ifx\csname CJK@cmaph/\f@family\CJK@plane\endcsname\relax
2897 \else\csname CJK@cmaph/\f@family\CJK@plane\endcsname\fi
2898 \else \CJK@addcmaph\CJK@plane \fi
2899 \else\xdef\font@name{%
2900 \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2901 \@gobble
2902 }%
2903 }{\@firstofone}%
```

This is the normal L<sup>A</sup>T<sub>E</sub>X definition.

```
2904 {\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.

```
2905 \ifx\pickup@font\MT@orig@pickupfont \else
2906 \MT@warning@nl{%
2907 Command \string\pickup@font\space is not defined as expected.%
2908 \MessageBreak Patching it anyway. Some things may break%
2909 (package)
2910 .\MessageBreak Double-check whether micro-typography is indeed%
2911 \MessageBreak applied to the document.%
2912 \MessageBreak (Hint: Turn on `verbose' mode)%
```

```

2913 </package>
2914 }%
2915 \fi

```

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

```
2916 \g@addto@macro\pickup@font{\begingroup}%
```

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

```

2917 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
2918 \g@addto@macro\pickup@font{%
2919   \escapechar\m@ne
2920 <*package>
2921 <debug> \global\MT@inannottrue
2922 <debug> \MT@glet\MT@pdf@annot\@empty
2923 <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.

```

2924 \MT@let@cn\MT@font\MT@subst@expandafter\string\font@name}%
2925 \ifx\MT@font\relax
2926   \let\MT@font\font@name
2927 \else
2928   \ifx\MT@font\font@name \else
2929 <debug> \MT@addto@annot{= substituted with \MT@font}%
2930   \MT@register@subst@font
2931   \fi
2932 \fi
2933 \MT@setupfont
2934 </package>
2935 <letterspace> \MT@tracking
2936 \endgroup
2937 }%
2938 <*package>

```

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

```

\MT@MT@pickupfont
\MT@ltx@pickupfont 2939 \let\MT@pickupfont\pickup@font
2940 \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
2941 \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.

```

2942 \g@addto@macro\do@subst@correction
2943 { \edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2944   \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 \MT@orig@add@accent would break the accent if different fonts are used for the base character and the accent. Fortunately, L<sup>A</sup>T<sub>E</sub>X takes care that the fonts used for the \accent are already set up, so that we cannot be overlooking them.

```

2945 \let\MT@orig@add@accent\add@accent
2946 \def\add@accent#1#2{%
2947   \MT@ltx@pickupfont
2948   \MT@orig@add@accent{#1}{#2}%
2949   \MT@MT@pickupfont

```

```

2950 }%
2951 </package>
2952 }
2953 <plain>\relax
2954 <*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.

```
2955 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

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

```
2956 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}
```

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

```

2957 \def\MT@register@subst@font{%
2958   \MT@exp@one@n\MT@in@clist\font@name\MT@font@list
2959   \ifMT@inlist@else
2960     \xdef\MT@font@list{\MT@font@list\font@name,}%
2961     \expandafter\MT@rem@from@clist\MT@font\MT@font@list
2962   \fi
2963 }

```

#### 14.2.10 Context-sensitive setup

Here are the variants for context-sensitive setup.

`\MT@active@features` The activated features are stored in this command.

```
2964 \let\MT@active@features\@empty
```

`\MT@check@font@cx` Every feature has its own list of fonts that have already been dealt with. If the font needn't be set up for a feature, we temporarily disable the corresponding setup command. This should be more efficient than book-keeping the fonts in lists associated with the combination of contexts, as we've done it before.

```

2965 \def\MT@check@font@cx{%
2966   \MT@if@true
2967   \MT@map@clist@c\MT@active@features{%
2968     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2969     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2970     \ifMT@inlist@
2971       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2972     \else
2973       \MT@if@false
2974     \fi
2975   }%
2976   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2977 }

```

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

```

2978 \def\MT@register@subst@font@cx{%
2979   \MT@map@clist@c\MT@active@features{%
2980     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
2981     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2982     \ifMT@inlist@ \else
2983       \MT@exp@cs\MT@xadd
2984       {MT@##1@\csname MT@##1@context\endcsname font@list}%
2985       {\font@name,}%
2986     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font
2987     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2988     \fi
2989   }%

```

2990 }

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

```
2991 \def\MT@register@font@cx{%
2992   \MT@map@clist@c\MT@active@features{%
2993     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2994       \MT@exp@cs\MT@xadd
2995         {MT@##1@c\sname MT@##1@context\endcsname font@list}%
2996       {\MT@font,}%
2997     \def\@tempa{##1}%
2998     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2999   \fi
3000 }%
3001 }
```

`\MT@maybe@rem@from@list` Recurse through all context font lists of the document and remove the font, unless it's the current context.

```
3002 \def\MT@maybe@rem@from@list#1{%
3003   \MT@ifstreql\@tempa/#1{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
3004     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
3005       \MT@font \csname MT@\@tempa @#1font@list\endcsname
3006   }%
3007 }
```

`\microtypecontext` The user may change the context, so that different setups are possible. This is especially useful for multi-lingual documents.

`\MT@microtypecontext`

Inside the preamble, this command shouldn't actually do anything but remember itself for later.

```
3008 \def\microtypecontext{\MT@begin@catcodes\MT@microtypecontext}
3009 \def\MT@microtypecontext#1{\MT@end@catcodes\MT@addto@setup{\microtypecontext{#1}}}
3010 \MT@addto@setup{%
3011   \DeclareRobustCommand\microtypecontext{%
3012     \MT@begin@catcodes
3013     \MT@microtypecontext
3014   }%
3015   \def\MT@microtypecontext#1{%
3016     \MT@end@catcodes
3017     \MT@setup@contexts
3018     \let\MT@reset@context\relax
```

We need to ensure that math fonts are set up anew.

```
3019   \MT@gl@et\glb@currsiz@empty
3020   \setkeys{MTC}{#1}%
3021   \selectfont
3022   \MT@reset@context
3023 }%
3024 }
```

`\textmicrotypecontext` This is just a wrapper around `\microtypecontext`.

```
\MT@textmicrotypecontext 3025 \DeclareRobustCommand\textmicrotypecontext{\MT@begin@catcodes\MT@textmicrotypecontext}
\MT@text@microtypecontext 3026 \def\MT@textmicrotypecontext#1{\MT@end@catcodes\MT@text@microtypecontext{#1}}
3027 \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 change.

`\MT@reset@context@`

```
3028 \def\MT@reset@context@{%
3029   \MT@vinfo{<<< Resetting contexts\on@line
3030   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
3031   <debug> \MT@tr@context/\MT@kn@context/\MT@sp@context
3032 }%
3033 \selectfont
3034 }
```

`\MT@setup@contexts` The first time `\microtypecontext` is called, we initialise the context lists and redefine the commands used in `\pickup@font`.

```

3035 \def\MT@setup@contexts{%
3036   \MT@map@clist@c\MT@active@features
3037   {\MT@gl@et@c\MT@##1@font@list}\MT@font@list}%
3038   \MT@gl@et\MT@check@font\MT@check@font@cx
3039   \MT@gl@et\MT@register@font\MT@register@font@cx
3040   \MT@gl@et\MT@register@subst@font\MT@register@subst@font@cx
3041   \MT@gl@et\MT@setup@contexts\relax
3042 }

```

Define context keys.

```

3043 \MT@map@clist@c\MT@features@long{%
3044   \define@key{MTC}{#1}[]{}%
3045   \edef\@tempb{\@nameuse{MT@rbba@#1}}%
3046   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
3047   \ifMT@inlist@

```

Using an empty context is only asking for trouble, therefore we choose the ‘@’ instead (hoping for the L<sup>A</sup>T<sub>E</sub>X users’ natural awe of this character).

```

3048   \MT@ifempty{#1}{\def\MT@val{}}{\def\MT@val{#1}}%
3049   \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
3050   <debug> \MT@info{1}{>>> no change of #1 context: `~\MT@val'}%
3051   \else
3052     \MT@vinfo{>>> Changing #1 context to `~\MT@val'\MessageBreak\on@line
3053     <debug> \space(previous: `~\@nameuse{MT@\@tempb @context}')}%
3054     }%
3055   \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

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

```

3056   \MT@gl@et@nn{MT@reset@\@tempb @codes}\MT@reset@\@tempb @codes@}%

```

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

```

3057   \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
3058   \MT@val \cname MT@\@tempb @doc@contexts\endcsname
3059   \ifMT@inlist@ \else
3060     \MT@exp@cs\MT@add{MT@\@tempb @doc@contexts}{\MT@val}%
3061   <debug> \MT@info{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
3062   \fi
3063   \MT@edef@n{MT@\@tempb @context}{\MT@val}%
3064   \fi
3065   \fi
3066 }%
3067 }

```

We also allow the activate shortcut.

```

3068 \define@key{MTC}{activate}[]{}%
3069 \setkeys{MTC}{protrusion={#1}}%
3070 \setkeys{MTC}{expansion={#1}}%
3071 }

```

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

```

\MT@ex@context 3072 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}%
\MT@tr@context 3073 \MT@def@n{MT@#1@context}{@}%
\MT@sp@context 3074 \MT@def@n{MT@#1@doc@contexts}{\@}%
\MT@kn@context 3075 }
\MT@kn@context 3076 \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`

`\MT@extra@context`

## 14.3 Configuration



### 14.3.1 Font sets

`\DeclareMicrotypeSet` 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.

```

3077 \def\DeclareMicrotypeSet{%
3078   \MT@begin@catcodes
3079   \ifstar
3080     \MT@DeclareSetAndUseIt
3081     \MT@DeclareSet
3082 }

\MT@DeclareSet

3083 \newcommand\MT@DeclareSet[3][]{%
3084   \MT@ifempty{#1}{%
3085     \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
3086   }{%
3087     \MT@map@clist@n{#1}{%
3088       \MT@ifempty{##1}\relax{%
3089         \MT@is@feature{##1}{set declaration `#2'}{%
3090           \MT@exp@one@n\MT@declare@sets
3091             {\csname MT@rbba@##1\endcsname}{#2}{#3}%
3092         }%
3093       }%
3094     }%
3095   }%
3096   \MT@end@catcodes
3097 }
```

```

\MT@DeclareSetAndUseIt

3098 \newcommand\MT@DeclareSetAndUseIt[3][]{%
3099   \MT@DeclareSet{#1}{#2}{#3}%
3100   \UseMicrotypeSet{#1}{#2}%
3101 }
```

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

```
3102 \let\MT@curr@set@name\empty
```

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

```

3103 \def\MT@declare@sets#1#2#3{%
3104   \def\MT@curr@set@name{#2}%
3105   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
3106     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
3107     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
3108       \MT@glet@nc{MT@#1list@##1@MT@curr@set@name}\@undefined
3109     }%
3110   }%
3111   \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
3112   <debug>\MT@edinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
3113   \setkeys{MT@#1@set}{#3}%
3114 }
```

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

```

3115 \def\MT@define@set@key@#1#2{%
3116   \define@key{MT@#2@set}{#1}[]{%
3117     \MT@glet@nc{MT@#2list@#1@MT@curr@set@name}\@empty
3118     \MT@map@clist@n{##1}{%
3119       \KV@esp@def\MT@val{###1}%
3120       \MT@get@highlevel{#1}%

```

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

```
3121 \MT@exp@two@n@g@addto@macro
3122 {\csname MT@#2list@#1@MT@curr@set@name\expandafter\endcsname}%
3123 {\MT@val,}%
3124 }%
```

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

```
3125 \expandafter\g@addto@macro\expandafter\MT@font@sets
3126 \csname MT@#2list@#1@MT@curr@set@name\endcsname
3127 <debug>\MT@edinfo@n1{1}{-- #1: \@nameuse{MT@#2list@#1@MT@curr@set@name}}%
3128 }%
3129 }
```

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

```
3130 \def\MT@get@highlevel#1{%
3131 \expandafter\MT@test@ast\MT@val*\@nil\relax{%
```

And ‘family = \*’ will become \familydefault.

```
3132 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
```

Test whether the command is actually defined.

```
3133 \MT@ifdefined@nTF{\@tempa default}%
3134 {\edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}}%
3135 {\MT@warning{\@backslashchar\@tempa default' is not a defined command.\MessageBreak
3136 Ignoring `#1 = {\@tempa*}' in font set\MessageBreak\MT@curr@set@name'}}%
3137 \let\MT@val\@empty}%
```

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

```
3138 }%
3139 }
```

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

```
3140 \def\MT@test@ast#1*#2\@nil{%
3141 \def\@tempa{#1}%
3142 \MT@ifempty{#2}%
3143 }
```

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

```
3144 \let\MT@font@sets\@empty
3145 \def\MT@fix@font@set#1{%
3146 \MT@ifdefined@c@T{#1}{%
3147 \xdef#1{#1}%
3148 \ifMT@fontspec
3149 \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
3150 \fi
3151 \global\@onelevel@sanitize#1%
3152 }%
3153 }
```

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

```
3154 \def\MT@define@set@key@size#1{%
3155 \define@key{MT@#1@set}{size}[]{}%
3156 \MT@map@clist@n{##1}{%
3157 \def\MT@val{###1}%
3158 \expandafter\MT@get@range\MT@val--\@nil
3159 \ifx\MT@val\relax\else
3160 \MT@exp@cs\MT@xadd
3161 {MT@#1list@size@MT@curr@set@name}%
3162 {{{\MT@lower}{\MT@upper}\relax}}%
3163 \fi
```

```

3164 }%
3165 <debug>\MT@info{n1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
3166 }%
3167 }

```

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  $\{\langle lower\ bound \rangle\}\{\langle upper\ bound \rangle\}\{\langle list\ name \rangle\}$ .

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

```

\MT@lower 3168 \def\MT@get@range#1-#2-#3\@nil{%
3169   \MT@ifempty{#1}{%
3170     \MT@ifempty{#2}{%
3171       \let\MT@val\relax
3172     }%
3173     \def\MT@lower{0}%
3174     \def\MT@val{#2}%
3175     \MT@get@size
3176     \edef\MT@upper{\MT@val}%
3177   }%
3178 }{%
3179   \def\MT@val{#1}%
3180   \MT@get@size
3181   \ifx\MT@val\relax \else
3182     \edef\MT@lower{\MT@val}%
3183     \MT@ifempty{#2}{%
3184       \MT@ifempty{#3}%
3185     }{\def\MT@upper{-1}}%

```

2048 pt is TeX's maximum font size.

```

3186   {\def\MT@upper{2048}}%
3187 }{%
3188   \def\MT@val{#2}%
3189   \MT@get@size
3190   \ifx\MT@val\relax \else
3191     \MT@ifdim\MT@lower>\MT@val{%
3192       \MT@error{%
3193         Invalid size range (\MT@lower\space > \MT@val) in font set
3194         '\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
3195     \edef\MT@upper{\MT@lower}%
3196     \edef\MT@lower{\MT@val}%
3197   }{%
3198     \edef\MT@upper{\MT@val}%
3199   }%
3200   \MT@ifdim\MT@lower=\MT@upper
3201   {\def\MT@upper{-1}}%
3202   \relax
3203   \fi
3204 }%
3205 \fi
3206 }%
3207 }

```

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

```

3208 \def\MT@get@size{%

```

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

```

3209   \if*\MT@val\relax
3210     \def\@tempa{\normalsize}%
3211   \else

```

```

3212 \MT@let@cn\@tempa{\MT@val}%
3213 \fi
3214 \ifx\@tempa\relax\else
3215 \MT@get@size@
3216 \fi

```

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

```

3217 \MT@ifdimen\MT@val{%
3218 \@defaultunits\@tempdima\MT@val pt\relax\@nnil
3219 \edef\MT@val{\strip@pt\@tempdima}%
3220 }{%
3221 \MT@warning{Could not parse font size `~\MT@val'\MessageBreak
3222 in font set `~\MT@curr@set@name'}%
3223 \let\MT@val\relax
3224 }%
3225 }

```

\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).

```

3226 \def\MT@get@size@@{%
3227 \begingroup
3228 \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
3229 \@tempa\@nil
3230 }

```

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

```

3231 ^^X\@ifclassloaded{svjour3}{%
3232 ^^X \def\MT@get@size@{%
3233 ^^X \@tempcnta=currentiflevel
3234 ^^X \MT@get@size@@
3235 ^^X \loop
3236 ^^X \ifnum\numexprcurrentiflevel-1>\@tempcnta
3237 ^^X \csname fi\endcsname
3238 ^^X \repeat
3239 ^^X }%
3240 ^^X}{%
3241 \let\MT@get@size@\MT@get@size@@
3242 ^^X}

```

\MT@define@set@key@font

```

3243 \def\MT@define@set@key@font#1{%
3244 \define@key{MT@#1@set}{font}[]{}%
3245 \MT@glet@nc{MT@#1list@font@\MT@curr@set@name}\@empty
3246 \MT@map@clist@n{##1}{%
3247 \def\MT@val{####1}%
3248 \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/}}\relax
3249 \expandafter\MT@get@font\MT@val////\@nil
3250 \MT@exp@two@n@g@addto@macro
3251 {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
3252 {\MT@val,%}%
3253 }%
3254 \expandafter\g@addto@macro\expandafter\MT@font@sets
3255 \csname MT@#1list@font@\MT@curr@set@name\endcsname
3256 <debug>\MT@debuginfo@n1{1}{-- font: \nameuse{MT@#1list@font@\MT@curr@set@name}}%
3257 }%
3258 }

```

\MT@get@font Translate any asterisks.

```

3259 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%

```

```

3260 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
3261 \ifx\MT@val\relax\def\MT@val{0}\fi
3262 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
3263 \let\MT@val\@tempb
3264 }

```

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

```

3265 \def\MT@get@font@#1#2#3#4#5#6{%
3266 \let\@tempb\@empty
3267 \def\MT@temp{#1/#2/#3/#4/#5}%
3268 \MT@get@axis{encoding}{#1}%
3269 \MT@get@axis{family} {#2}%
3270 \MT@get@axis{series} {#3}%
3271 \MT@get@axis{shape} {#4}%
3272 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
3273 \MT@ifempty{#5}{%
3274 \MT@warn@axis@empty{size}{\string\normalsize}%
3275 \def\MT@val{*}%
3276 }{%
3277 \def\MT@val{#5}%
3278 }%
3279 \MT@get@size
3280 }

```

\MT@get@axis

```

3281 \def\MT@get@axis#1#2{%
3282 \def\MT@val{#2}%
3283 \MT@get@highlevel{#1}%
3284 \MT@ifempty\MT@val{%
3285 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
3286 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
3287 }\relax
3288 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
3289 }

```

\MT@warn@axis@empty

```

3290 \def\MT@warn@axis@empty#1#2{%
3291 \MT@warning{#1 axis is empty in font specification\MessageBreak
3292 ~\MT@temp'. Using ~#2' instead}%
3293 }

```

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

```

3294 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
3295 \MT@define@set@key@{encoding}{#1}%
3296 \MT@define@set@key@{family} {#1}%
3297 \MT@define@set@key@{series} {#1}%
3298 \MT@define@set@key@{shape} {#1}%
3299 \MT@define@set@key@size {#1}%
3300 \MT@define@set@key@font {#1}%
3301 }

```

\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.

```

3302 \def\UseMicrotypeSet{%
3303 \MT@begin@catcodes
3304 \MT@UseMicrotypeSet
3305 }

```

\MT@UseMicrotypeSet

```

3306 \newcommand*\MT@UseMicrotypeSet[2][]{%
3307 \MT@ifempty{#1}{%
3308 \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
3309 }{%

```

```

3310 \MT@map@clist@n{#1}{%
3311 \MT@ifempty{##1}\relax{%
3312 \MT@is@feature{##1}{activation of set `#2'}{%
3313 \MT@exp@one@n\MT@use@set
3314 {\csname MT@rbba@##1\endcsname}{#2}%
3315 }%
3316 }%
3317 }}%
3318 }%
3319 \MT@end@catcodes
3320 }

```

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

```

\MT@ex@setname 3321 \def\MT@use@set#1#2{%
\MT@tr@setname 3322 \MT@ifdefined@n@TF{MT@#1@set@#2}{%
3323 \MT@xdef@n{MT@#1@setname}{#2}%
\MT@sp@setname 3324 }{%
\MT@kn@setname 3325 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
\MT@use@set 3326 \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
3327 }%
3328 \MT@error{%
3329 The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
3330 Using set ` \@nameuse{MT@#1@setname}' instead{}}%
3331 }%
3332 }

```

\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.

```

3333 \def\DeclareMicrotypeSetDefault{%
3334 \MT@begin@catcodes
3335 \MT@DeclareMicrotypeSetDefault
3336 }

```

\MT@DeclareMicrotypeSetDefault

```

3337 \newcommand*\MT@DeclareMicrotypeSetDefault[2][ ]{%
3338 \MT@ifempty{#1}{%
3339 \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
3340 }{%
3341 \MT@map@clist@n{#1}{%
3342 \MT@ifempty{##1}\relax{%
3343 \MT@is@feature{##1}{declaration of default set `#2'}{%
3344 \MT@exp@one@n\MT@set@default@set
3345 {\csname MT@rbba@##1\endcsname}{#2}%
3346 }%
3347 }%
3348 }}%
3349 }%
3350 \MT@end@catcodes
3351 }

```

\MT@default@pr@set

```

\MT@default@ex@set 3352 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 3353 \MT@ifdefined@n@TF{MT@#1@set@#2}{%
3354 (debug) \MT@info{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
\MT@default@sp@set 3355 \MT@xdef@n{MT@default@#1@set}{#2}%
\MT@default@kn@set 3356 }{%
\MT@set@default@set 3357 \MT@error{%
3358 The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
3359 Cannot make it the default set. Using set\@nameuse{MT@all@#1@set} instead{}}%
3360 \MT@xdef@n{MT@default@#1@set}{all}%
3361 }%
3362 }

```

### 14.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version `\MT@variants` appends to the list.

```

3363 \let\MT@variants\@empty
3364 \def\DeclareMicrotypeVariants{%
3365   \MT@begin@catcodes
3366   \ifstar
3367     \MT@DeclareVariants
3368     {\let\MT@variants\@empty\MT@DeclareVariants}%
3369 }

```

`\MT@DeclareVariants`

```

3370 \def\MT@DeclareVariants#1{%
3371   \MT@map@clist@n{#1}%
3372   \def\@tempa{##1}%
3373   \@onelevel@sanitize\@tempa
3374   \xdef\MT@variants{\MT@variants{\@tempa}}%
3375 }%
3376 \MT@end@catcodes
3377 }

```

`\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.

```

3378 \def\DeclareMicrotypeAlias{%
3379   \MT@begin@catcodes
3380   \MT@DeclareMicrotypeAlias
3381 }

```

`\MT@DeclareMicrotypeAlias`

```

3382 \newcommand*\MT@DeclareMicrotypeAlias[2]{%
3383   \def\@tempb{#2}%
3384   \@onelevel@sanitize\@tempb
3385   \MT@ifdefined@n@T{MT@#1@alias}{%
3386     \MT@warning{Alias font family '\@tempb' will override
3387       alias '\@nameuse{MT@#1@alias}'\MessageBreak
3388       for font family '#1'}%
3389   \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.

```

3390   \MT@ifdefined@c@T{MT@family}{%
3391     \debug\MT@dinfo{1}{Activating alias font '\@tempb' for '\MT@family'}%
3392     \MT@gl@et{MT@familyalias}{\@tempb}
3393   }%
3394   \MT@end@catcodes
3395 }

```

`\LoadMicrotypeFile` May be used to load a configuration file manually.

```

3396 \def\LoadMicrotypeFile#1{%
3397   \edef\@tempa{\zap@space#1 \@empty}%
3398   \@onelevel@sanitize\@tempa
3399   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
3400   \ifMT@inlist@
3401     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
3402   \else
3403     \MT@xadd\MT@file@list{\@tempa,%}
3404     \MT@begin@catcodes
3405     \InputIfFileExists{mt-\@tempa.cfg}{%
3406       \edef\MT@curr@file{mt-\@tempa.cfg}%
3407       \MT@vinfo{... Loading configuration file \MT@curr@file}%
3408     }{%

```

```

3409     \MT@warning{Configuration file mt-\@tempa.cfg\MessageBreak
3410               does not exist}%
3411   }%
3412   \MT@end@catcodes
3413   \fi
3414 }
3415 \end{package}
3416 \end{package|letterspace}

```

### 14.3.3 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 3417 \pdfTeX-def|luatex-def
3418 \pdfTeX-def\MT@requires@pdftex5{
3419 \def\DisableLigatures{%
3420   \MT@begin@catcodes
3421   \MT@DisableLigatures
3422 }
3423 \newcommand*\MT@DisableLigatures[2][ ]{%
3424   \MT@ifempty{#1}\relax{\gdef\MT@n1@ligatures{#1}}%
3425   \xdef\MT@active@features{\MT@active@features,n1}%
3426   \global\MT@noligaturestrue
3427   \MT@declare@sets{n1}{no ligatures}{#2}%
3428   \gdef\MT@n1@setname{no ligatures}%
3429   \MT@end@catcodes
3430 }
3431 \pdfTeX-def}{
3432 \pdfTeX-def|luatex-def

```

If pdfTeX is too old, we throw an error.

```

3433 \pdfTeX-def|xetex-def
3434 \renewcommand*\DisableLigatures[2][ ]{%
3435   \MT@error{Disabling ligatures of a font is only possible\MessageBreak
3436     with pdftex version 1.30 or newer.\MessageBreak
3437     Ignoring \string\DisableLigatures}{%
3438     \pdfTeX-def Upgrade
3439     \xetex-def Use
3440     pdftex.}%
3441 }
3442 \pdfTeX-def}
3443 \pdfTeX-def|xetex-def

```

### 14.3.4 Interaction with babel

`\DeclareMicrotypeBabelHook` Declare the context that should be loaded when a babel language is selected. The command will not check whether a previous declaration will be overwritten.

```

3444 \begin{package}
3445 \def\DeclareMicrotypeBabelHook#1#2{%
3446   \MT@map@clist@n{#1}{%
3447     \KV@sp@def\@tempa{##1}%
3448     \MT@gdef@n{\MT@babel@{\@tempa}{#2}}%
3449   }%
3450 }
3451 \end{package}

```

### 14.3.5 Fine tuning

The commands `\SetExpansion` and `\SetProtrusion` provide an interface for setting the character protrusion resp. expansion factors for a set of fonts.



`\SetProtrusion` This macro accepts three arguments: [options,] set of font attributes and list of character protrusion factors.

A new macro called `\MT@pr@c@<name>` will be defined to be `<#3>` (i.e., the list of characters, not expanded).

```
3452 <*pdfTeX-def|xetex-def|luatex-def>
3453 \def\SetProtrusion{%
3454   \MT@begin@catcodes
3455   \MT@SetProtrusion
3456 }
```

`\MT@SetProtrusion` We want the catcodes to be correct even if this is called in the preamble.

```
\MT@pr@c@name 3457 \newcommand*\MT@SetProtrusion[3] [] {%
```

`\MT@extra@context` 3458 \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.

```
3459 \MT@set@named@keys{MT@pr@c}{#1}%
3460 <debug>\MT@info{1}{creating protrusion list `~\MT@pr@c@name'}%
3461 \def\MT@permuteList{pr@c}%
3462 \setkeys{MT@cfg}{#2}%
```

We have parsed the second argument, and can now define macros for all permutations of the font attributes to point to `\MT@pr@c@<name>`, ...

```
3463 \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.

```
3464 \MT@gdefn{MT@pr@c@MT@pr@c@name}{#3}%
3465 \MT@end@catcodes
3466 }
3467 </pdfTeX-def|xetex-def|luatex-def>
```

`\SetExpansion` `\SetExpansion` only differs in that it allows some extra options (stretch, shrink, step, auto).

```
3468 <*pdfTeX-def|luatex-def>
3469 \def\SetExpansion{%
3470   \MT@begin@catcodes
3471   \MT@SetExpansion
3472 }
```

`\MT@SetExpansion`

```
\MT@ex@c@name 3473 \newcommand*\MT@SetExpansion[3] [] {%
```

`\MT@extra@context` 3474 \let\MT@extra@context\@empty

`\MT@permuteList` 3475 \MT@set@named@keys{MT@ex@c}{#1}%

```
3476 \MT@ifdefinedn@T{MT@ex@c@MT@ex@c@name @factor}{%
3477   \ifnum\csname MT@ex@c@MT@ex@c@name @factor\endcsname > \@m
3478     \MT@warningn1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
3479       too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
3480       maximum of 1000}%
3481     \MT@gl@et@nc{MT@ex@c@MT@ex@c@name @factor}\@m
3482   \fi
3483 }%
```

```
3484 <debug>\MT@info{1}{creating expansion list `~\MT@ex@c@name'}%
3485 \def\MT@permuteList{ex@c}%
3486 \setkeys{MT@cfg}{#2}%
3487 \MT@permute
3488 \MT@gdefn{MT@ex@c@MT@ex@c@name}{#3}%
3489 \MT@end@catcodes
3490 }
```

`\SetTracking`

```
3491 \def\SetTracking{%
```

```

3492 \MT@begin@catcodes
3493 \MT@SetTracking
3494 }

```

\MT@SetTracking      Third argument may be empty.

```

3495 \newcommand*\MT@SetTracking[3] [] {%
3496   \let\MT@extra@context\@empty
3497   \MT@set@named@keys{MT@tr@{c}}{#1}%
3498   <debug>\MT@info{1}{creating tracking list `~\MT@tr@{c}@name'}%
3499   \def\MT@permute{list{tr@{c}}}%
3500   \setkeys{MT@cfg}{#2}%
3501   \MT@permute
3502   \KV@sp@def\@tempa{#3}%
3503   \MT@ifempty\@tempa\relax{%
3504     \MT@ifint\@tempa
3505     {\MT@xdef\MT@tr@{c}\MT@tr@{c}@name}{\@tempa}}%
3506     {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
3507       tracking set `~\MT@curr@set@name'}}}%
3508   \MT@end@catcodes
3509 }
3510 </pdfTeX-def|LaTeX-def>

```

\SetExtraSpacing

```

3511 <*pdfTeX-def>
3512 \def\SetExtraSpacing{%
3513   \MT@begin@catcodes
3514   \MT@SetExtraSpacing
3515 }

```

\MT@SetExtraSpacing

```

\MT@sp@{c}@name 3516 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 3517   \let\MT@extra@context\@empty
\MT@permute{list 3518   \MT@set@named@keys{MT@sp@{c}}{#1}%
3519   <debug>\MT@info{1}{creating spacing list `~\MT@sp@{c}@name'}%
3520   \def\MT@permute{list{sp@{c}}}%
3521   \setkeys{MT@cfg}{#2}%
3522   \MT@permute
3523   \MT@gdef\MT@sp@{c}\MT@sp@{c}@name}{#3}%
3524   \MT@end@catcodes
3525 }

```

\SetExtraKerning

```

3526 \def\SetExtraKerning{%
3527   \MT@begin@catcodes
3528   \MT@SetExtraKerning
3529 }

```

\MT@SetExtraKerning

```

\MT@kn@c@name 3530 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 3531   \let\MT@extra@context\@empty
\MT@permute{list 3532   \MT@set@named@keys{MT@kn@c}{#1}%
3533   <debug>\MT@info{1}{creating kerning list `~\MT@kn@c@name'}%
3534   \def\MT@permute{list{kn@c}}%
3535   \setkeys{MT@cfg}{#2}%
3536   \MT@permute
3537   \MT@gdef\MT@kn@c\MT@kn@c@name}{#3}%
3538   \MT@end@catcodes
3539 }
3540 </pdfTeX-def>

```

\MT@set@named@keys      We first set the name (if specified), then remove it from the list, and set the remaining keys.

\MT@options

```

3541 <*package>
3542 \def\MT@set@named@keys#1#2{%

```

```

3543 \def\x##1name=##2,##3\@nil{%
3544   \setkeys{#1}{name=##2}%
3545   \gdef\MT@options{##1##3}%
3546   \MT@rem@from@clist{name=}\MT@options
3547 }%
3548 \x#2,name=,\@nil
3549 \@expandtwoargs\setkeys{#1}\MT@options
3550 }

```

`\MT@define@code@key` Define the keys for the configuration lists (which are setting the codes, in pdfTeX speak).

```

3551 \def\MT@define@code@key#1#2{%
3552   \define@key{MT@#2}{#1}[]{}%
3553   \@tempcnta=\@ne
3554   \MT@map@cliston{##1}%
3555   \KV@sp@def\MT@val{###1}%

```

Here, too, we allow for something like ‘bf\*’. It will be expanded immediately.

```

3556   \MT@get@highlevel{#1}%
3557   \MT@edefon{MT@temp#1\the\@tempcnta}{\MT@val}%
3558   \advance\@tempcnta \@ne
3559 }%
3560 }%
3561 }

```

`\MT@define@code@key@family` Remove fontspec’s internal feature counter.

```

3562 \def\MT@define@code@key@family#1{%
3563   \define@key{MT@#1}{family}[]{}%
3564   \@tempcnta=\@ne
3565   \MT@map@cliston{##1}%
3566   \KV@sp@def\MT@val{###1}%
3567   \MT@get@highlevel{family}%
3568   \ifMT@fontspec
3569     \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
3570     \fi
3571     \MT@edefon{MT@tempfamily\the\@tempcnta}{\MT@val}%
3572     \advance\@tempcnta \@ne
3573   }%
3574 }%
3575 }

```

`\MT@define@code@key@size` `\MT@tempsize` must be in a `\csname`, so that it is at least `\relax`, not undefined.

```

3576 \def\MT@define@code@key@size#1{%
3577   \define@key{MT@#1}{size}[]{}%
3578   \MT@map@cliston{##1}%
3579   \KV@sp@def\MT@val{###1}%
3580   \expandafter\MT@get@range\MT@val--\@nil
3581   \ifx\MT@val\relax \else
3582     \MT@exp@cs\MT@xadd{MT@tempsize}%
3583     {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
3584   \fi
3585 }%
3586 }%
3587 }

```

`\MT@define@code@key@font`

```

3588 \def\MT@define@code@key@font#1{%
3589   \define@key{MT@#1}{font}[]{}%
3590   \MT@map@cliston{##1}%
3591   \KV@sp@def\MT@val{###1}%
3592   \MT@ifstreq\MT@val{*/*/*/*/*}\relax
3593   \expandafter\MT@get@font@and@size\MT@val////\@nil
3594   \ifMT@fontspec
3595     \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
3596     \fi

```

```

3597 \MT@xdefn{MT@MT@permutelist @\@tempb\MT@extra@context}%
3598 {\csname MT@MT@permutelist @name\endcsname}%
3599 (debug)\MT@edinfo{1}{initialising: use list for font \@tempb=MT@val
3600 (debug) \ifx\MT@extra@context\@empty\else\MessageBreak
3601 (debug) (context: \MT@extra@context)\fi}%
3602 \MT@exp@cs\MT@xaddb
3603 {MT@MT@permutelist @\@tempb\MT@extra@context @sizes}%
3604 {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
3605 }%
3606 }%
3607 }

```

`\MT@get@font@and@size` Translate any asterisks and split off the size.

```

3608 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
3609 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
3610 }

3611 \MT@define@code@key{encoding}{cfg}
3612 \MT@define@code@key{family} {cfg}
3613 \MT@define@code@key{series} {cfg}
3614 \MT@define@code@key{shape} {cfg}
3615 \MT@define@code@key{size} {cfg}
3616 \MT@define@code@key{font} {cfg}

```

`\MT@define@opt@key`

```

3617 \def\MT@define@opt@key#1#2{%
3618 \define@key{MT@#1c}{#2}[]{\MT@ifempty{##1}\relax{%
3619 \MT@xdefn{MT@#1c@MT@curr@set@name @#2}{##1}}}%
3620 }

```

`\MT@listname@count` The options in the optional first argument.

```

3621 \newcount\MT@listname@count
3622 \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`).

```

3623 \define@key{MT@#1c}{name}[]{%
3624 \MT@ifempty{##1}{%
3625 \MT@ifdefined@n@TF{MT@#1c@MT@curr@file/\the\inputlineno}{%
3626 \global\advance\MT@listname@count@ne
3627 \MT@edefn{MT@#1c@name}{\MT@curr@file/\the\inputlineno
3628 (\number\MT@listname@count)}%
3629 }%
3630 \MT@edefn{MT@#1c@name}{\MT@curr@file/\the\inputlineno}%
3631 }%
3632 }%
3633 \MT@edefn{MT@#1c@name}{##1}%
3634 \MT@ifdefined@n@T{MT@#1c@\csname MT@#1c@name\endcsname}{%
3635 \MT@warning{Redefining \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1c@name}}%
3636 }%
3637 }%
3638 \MT@let@cn\MT@curr@set@name{MT@#1c@name}%
3639 }%
3640 \MT@define@opt@key{#1}{load}%
3641 \MT@define@opt@key{#1}{factor}%
3642 \MT@define@opt@key{#1}{preset}%
3643 \MT@define@opt@key{#1}{inputenc}%

```

Only one context is allowed. This might change in the future.

```

3644 \define@key{MT@#1c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
3645 }
3646 </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.

```

3647 (*pdfTeX-def|luatex-def)
3648 (pdfTeX-def)\MT@requires@pdfTeX7{
3649   \define@key{MT@ex@c}{context}[]{}%
3650   \MT@ifempty{#1}\relax{%
3651     \MT@gllet\MT@copy@font\MT@copy@font@
3652     \def\MT@extra@context{#1}%
3653   }%
3654 }
3655 \MT@addto@setup{%
3656   \define@key{MT@ex@c}{context}[]{}%
3657   \ifx\MT@copy@font\MT@copy@font@
3658     \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3659   \else
3660     \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
3661       Ignoring `context' key\on@line}%
3662     {Either move the settings inside the preamble,\MessageBreak
3663       or load the package with the `copyfonts' option.}%
3664   \fi
3665 }%
3666 }

```

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.

```

3667   \define@key{MT@pr@c}{context}[]{}%
3668   \MT@ifempty{#1}\relax{%
3669     \MT@gllet\MT@copy@font\MT@copy@font@
3670     \def\MT@extra@context{#1}%
3671   }%
3672 }
3673 \MT@addto@setup{%
3674   \define@key{MT@pr@c}{context}[]{}%
3675   \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3676   \ifx\MT@copy@font\MT@copy@font@\else
3677     \MT@warning{\If protrusion contexts don't work as expected,
3678       \MessageBreak load the package with the `copyfonts' option}%
3679   \fi
3680 }%
3681 }
3682 (/pdfTeX-def|luatex-def)
3683 (*pdfTeX-def)
3684 {}{
3685   \define@key{MT@ex@c}{context}[]{}%
3686   \MT@error{Expansion contexts only work with pdfTeX 1.40.4\MessageBreak
3687     or later. Ignoring `context' key\on@line}%
3688   {Upgrade pdfTeX.}%
3689 }
3690 (/pdfTeX-def)
3691 (*pdfTeX-def|xetex-def)
3692   \define@key{MT@pr@c}{context}[]{}%
3693   \MT@error{Protrusion contexts only work with pdfTeX
3694     1.40.4\MessageBreak or later.
3695     \MessageBreak or luatex.
3696     Ignoring `context' key\on@line}%
3697   {Upgrade pdfTeX.}%
3698   {Use pdfTeX or luatex.}%
3699 }
3700 (/pdfTeX-def|xetex-def)

```

```

3701 <pdfTeX-def>
\MT@warn@nodim
3702 <*package>
3703 \def\MT@warn@nodim#1{%
3704   \MT@warning{'\@tempa' is not a dimension.\MessageBreak
3705             Ignoring it and setting values relative to\MessageBreak #1}%
3706 }
3707 </package>

```

Protrusion codes may be relative to character width, or to any dimension.

```

3708 <*pdfTeX-def|xetex-def|luatex-def>
3709 \define@key{MT@pr@c}{unit}[Character]{%
3710   \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
3711   \def\@tempa{#1}%
3712   \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.

```

3713     \MT@ifdimen\@tempa
3714     {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
3715     {\MT@warn@nodim{character widths}}}%
3716   }%
3717 }
3718 </pdfTeX-def|xetex-def|luatex-def>

```

Tracking may only be relative to a dimension.

```

3719 <*pdfTeX-def|luatex-def>
3720 \define@key{MT@tr@c}{unit}[1em]{%
3721   \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
3722   \def\@tempa{#1}%
3723   \MT@ifdimen\@tempa
3724   {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
3725   {\MT@warn@nodim{1em}%
3726    \MT@gdef@n{MT@tr@c@MT@curr@set@name @unit}{1em}}%
3727 }
3728 </pdfTeX-def|luatex-def>

```

Spacing and kerning codes may additionally be relative to space dimensions.

```

3729 <*pdfTeX-def>
3730 \MT@map@clist@n{sp,kn}{%
3731   \define@key{MT@#1@c}{unit}[space]{%
3732     \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
3733     \def\@tempa{##1}%
3734     \MT@ifstreq\@tempa{character}\relax{%
3735       \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3736       \MT@ifstreq\@tempa{space}\relax{%
3737         \MT@ifdimen\@tempa
3738         {\MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3739         {\MT@warn@nodim{width of space}}}%
3740       }%
3741     }%
3742   }%
3743 }
3744 </pdfTeX-def>

```

The first argument to \SetExpansion accepts some more options.

```

3745 <*pdfTeX-def|luatex-def>
3746 \MT@map@clist@n{stretch,shrink,step}{%
3747   \define@key{MT@ex@c}{#1}[]{%
3748     \MT@ifempty{##1}\relax{%
3749       \MT@ifint{##1}{%

```

A space terminates the number.

```

3750     \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%

```

```

3751     }{%
3752     \MT@warning{%
3753     Value `##1' for option `#1' is not a number.\MessageBreak
3754     Ignoring it}%
3755     }%
3756   }%
3757 }%
3758 }
3759 \define@key{MT@ex@}{auto}[true]{%
3760   \def\@tempa{#1}%
3761   \csname if\@tempa\endcsname

Don't use autoexpand for pdfTeX version older than 1.20.
3762 \pdfTeX-def \MT@requires@pdftex4%
3763 \luatex-def \MT@requires@luatex3\relax
3764 {\MT@gdefn{MT@ex@}{MT@curr@set@name @auto}{autoexpand}}%
3765 \pdfTeX-def \MT@warning{pdftex too old for automatic font expansion}}%
3766 \else
3767 \pdfTeX-def \MT@requires@pdftex4%
3768 \luatex-def
3769 \MT@requires@luatex3{%
3770 \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
3771 \luatex}}%
3772 \luatex-def
3773 {\MT@gletnc{MT@ex@}{MT@curr@set@name @auto}\@empty}%
3774 \pdfTeX-def \relax
3775 \fi
3776 }

```

Tracking: Interword spacing and outer kerning. The variant with space just in case `\SetTracking` is called inside an argument (e.g., to `\IfFileExists`).

```

3777 \MT@define@opt@key{tr}{spacing}
3778 \MT@define@opt@key{tr}{outerspacing}
3779 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3780 \define@key{MT@tr@}{noligatures}[]%
3781 {\MT@xdefn{MT@tr@}{MT@curr@set@name @noligatures}{#1}}
3782 \define@key{MT@tr@}{outer spacing}[]{\setkeys{MT@tr@}{outerspacing={#1}}}
3783 \define@key{MT@tr@}{outer kerning}[]{\setkeys{MT@tr@}{outerkerning={#1}}}
3784 \define@key{MT@tr@}{no ligatures}[]{\setkeys{MT@tr@}{noligatures={#1}}}
3785 \pdfTeX-def\luatex-def

```

### 14.3.6 Character inheritance

`\DeclareCharacterInheritance` This macro may be used in the configuration files to declare characters that should inherit protrusion resp. expansion values from other characters. Thus, there is no need to define all accented characters (e.g., `\'a`, `\'a`, `\^a`, `\~a`, `\"a`, `\r{a}`, `\k{a}`, `\u{a}`), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

`\MT@inh@feat` The optional argument may be used to restrict the list to some features,  
`\MT@extra@inputenc` and to specify an input encoding.

```

3786 \package
3787 \renewcommand*\DeclareCharacterInheritance[1][]{%
3788   \let\MT@extra@context\@empty
3789   \let\MT@extra@inputenc\@undefined
3790   \let\MT@inh@feat\@empty
3791   \setkeys{MT@inh@}{#1}%
3792   \MT@begin@catcodes
3793   \MT@set@inh@list

```

3794 }

\MT@set@inh@list No need to create an inheritance list for tracking.

```

3795 \def\MT@set@inh@list#1#2{%
3796   \MT@ifempty\MT@inh@feat{%
3797     \MT@map@clist@c\MT@features{%
3798       \MT@ifstreq{#1}{tr}\relax{\MT@declare@char@inh{#1}{#2}}%
3799     }%
3800   }%
3801   \MT@map@clist@c\MT@inh@feat{%
3802     \KV@sp@def\@tempa{#1}%
3803     \MT@ifempty\@tempa\relax{%
3804       \edef\@tempa{\csname MT@rbba@\@tempa\endcsname}%
3805       \MT@ifstreq\@tempa{tr}\relax{%
3806         \MT@exp@one@n\MT@declare@char@inh{\@tempa}{#1}{#2}}%
3807     }%
3808   }%
3809   \MT@end@catcodes
3810 }
```

The keys for the optional argument.

```

3811 \MT@map@clist@c\MT@features@long{%
3812   \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}%
3813   \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

\MT@declare@char@inh The lists cannot be given a name by the user.

```

3814 \def\MT@declare@char@inh#1#2#3{%
3815   \MT@edefn{MT@#1@inh@name}%
3816     {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3817   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3818   \MT@ifdefined@c@T\MT@extra@inputenc{%
3819     \MT@xdefn{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3820   (debug)\MT@dinfo{1}{creating inheritance list ~\@nameuse{MT@#1@inh@name}}%
3821   \MT@gdefn{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3822   \def\MT@permutelist{#1@inh}%
3823   \setkeys{MT@inh}{#2}%
3824   \MT@permute
3825 }
```

Parse the second argument. \DeclareCharacterInheritance may also be set up for various combinations. We can reuse the key setup from the configuration lists (\Set...).

```

3826 \MT@define@code@key{encoding}{inh}
3827 \MT@define@code@key{family} {inh}
3828 \MT@define@code@key{series} {inh}
3829 \MT@define@code@key{shape} {inh}
3830 \MT@define@code@key{size} {inh}
3831 \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).

```

3832 \def\MT@inh@do#1,{%
3833   \ifx\relax#1\empty \else
3834     \MT@inh@split #1==\relax
3835     \expandafter\MT@inh@do
3836   \fi
3837 }
```

\MT@inh@split Only gather the inheriting characters here. Their codes will actually be set in



```

\MT@set@{feature}@codes.
3838 </package>
3839 <*pdfTeX-def|xetex-def|luatex-def>
3840 \def\MT@inh@split#1=#2=#3\relax{%
3841   \def\@tempa{#1}%
3842   \ifx\@tempa\empty \else
3843     \MT@get@slot
3844     <pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3845     <xetex-def> \ifx\MT@char\empty\else
3846       \let\MT@val\MT@char
3847       \MT@map@clist@n{#2}{%
3848         \def\@tempa{##1}%
3849         \ifx\@tempa\empty \else
3850           \MT@get@slot
3851           <pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3852           <xetex-def> \ifx\MT@char\empty\else
3853             \MT@exp@cs\MT@xadd\MT@inh@MT@listname @\MT@val @{{\MT@char}}%
3854             \fi
3855             \fi
3856             }%
3857 <debug>\MT@edinfo@n1{2}{children of #1 (\MT@val):
3858 <debug> \nameuse{MT@inh@MT@listname @\MT@val @}}%
3859 \fi
3860 \fi
3861 }
3862 </pdfTeX-def|xetex-def|luatex-def>

```

### 14.3.7 Permutation

\MT@permute      Calling \MT@permute will define commands for all permutations of the specified font  
\MT@permute@    attributes of the form \MT@{list type}@/{encoding}/{family}/{series}/{shape}/{| \*} to  
\MT@permute@@   be the expansion of \MT@{list type}@name, i.e., the name of the currently defined list.  
\MT@permute@@@   Size ranges are held in a separate macro called \MT@{list type}@/{font axes}@sizes,  
\MT@permute@@@@   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:

```

3863 <*package>
3864 \MT@gdef@n{\MT@pr@c@U/euroitc///}{euroitc}
3865 \MT@gdef@n{\MT@pr@c@U/euroitcs///}{euroitc}
3866 \MT@gdef@n{\MT@pr@c@U/euroitc//it/}{euroitci}
3867 \MT@gdef@n{\MT@pr@c@U/euroitcs//it/}{euroitci}
3868 \MT@gdef@n{\MT@pr@c@euroitc}{E={100,50}}
3869 \MT@gdef@n{\MT@pr@c@euroitci}{E={100,}}
3870 \def\MT@permute{%
3871   \let\MT@cnt@encoding\@ne
3872   \MT@permute@
3873   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3874   \MT@glet\MT@temp@size\undefined
3875 }
3876 \def\MT@permute@{%
3877   \let\MT@cnt@family\@ne

```

Undefine commands for the next round.

```

3878 \MT@permute@@
3879 \MT@increment\MT@cnt@encoding
3880 \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
3881 \MT@permute@
3882 }
3883 \def\MT@permute@@@{%
3884 \let\MT@cnt@series\@ne
3885 \MT@permute@@@
3886 \MT@increment\MT@cnt@family
3887 \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3888 \MT@permute@@
3889 }
3890 \def\MT@permute@@@{%
3891 \let\MT@cnt@shape\@ne
3892 \MT@permute@@@
3893 \MT@increment\MT@cnt@series
3894 \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3895 \MT@permute@@@
3896 }
3897 \def\MT@permute@@@@{%
3898 \MT@permute@@@@
3899 \MT@increment\MT@cnt@shape
3900 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3901 \MT@permute@@@@
3902 }

```

\MT@permute@@@@ In order to save some memory, we can ignore unused encodings (inside the document).

```

3903 \def\MT@permute@@@@@{%
3904 \MT@permute@define{encoding}%
3905 \ifMT@document
3906 \ifx\MT@tempencoding\@empty \else
3907 \MT@ifdefined@n@TF{T@MT@tempencoding}\relax
3908 {\expandafter\expandafter\expandafter\@gobble}%
3909 \fi
3910 \fi
3911 \MT@permute@@@@@
3912 }

```

\MT@permute@@@@@

```

3913 \def\MT@permute@@@@@{%
3914 \MT@permute@define{family}%
3915 \MT@permute@define{series}%
3916 \MT@permute@define{shape}%
3917 \edef\@tempa{\MT@tempencoding
3918 \MT@tempfamily
3919 \MT@tempseries
3920 \MT@tempshape
3921 \MT@ifdefined@c@T\MT@tempsize *}

```

Some sanity checks: an encoding must be specified (unless nothing else is).

```

3922 \MT@ifstreq\@tempa{////}\relax{%
3923 \ifx\MT@tempencoding\@empty
3924 \MT@warning{%
3925 You have to specify an encoding for\MessageBreak
3926 \@nameuse{MT@abbr@MT@permutelist} list
3927 ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3928 Ignoring it}%
3929 \else
3930 \MT@ifdefined@c@TF\MT@tempsize{%

```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```

3931 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3932 \MT@map@tlist@c\MT@tempsize\MT@check@rlist

```

```

3933 }%
3934 \MT@exp@cs\MT@xaddb
3935 {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3936 \MT@tempsize
3937 <debug>\MT@info@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3938 <debug> sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3939 <debug> @sizes\endcsname}%
3940 }{%

```

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.

```

3941 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3942 \MT@ifstreq{\csname MT@MT@permutelist @\@tempa\MT@extra@context\endcsname}%
3943 {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
3944 \relax}%
3945 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3946 ~\@nameuse{MT@MT@permutelist @name}' will\MessageBreak override
3947 list ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3948 for \MessageBreak font ~\@tempa'}%
3949 }%
3950 }%
3951 <debug>\MT@info@n1{1}{initialising: use list for font \@tempa
3952 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
3953 <debug> (context: \MT@extra@context)\fi}%
3954 }%
3955 \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
3956 {\csname MT@MT@permutelist @name\endcsname}%
3957 \fi
3958 }%
3959 }

```

\MT@permute@define Define the commands.

```

3960 \def\MT@permute@define#1{%
3961 \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3962 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3963 {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3964 {\MT@let@nc{MT@temp#1}\@empty}%
3965 }

```

\MT@permute@reset Reset the commands.

```

3966 \def\MT@permute@reset#1{%
3967 \@tempcnta=\@ne
3968 \MT@loop
3969 \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3970 \advance\@tempcnta\@ne
3971 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3972 \iftrue
3973 \iffalse
3974 \MT@repeat
3975 }

```

\MT@check@rlist For every new range item in \MT@tempsize, check whether it overlaps with ranges in the existing list.

```

3976 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}

```

\MT@check@rlist@ Define the current new range and ...

```

3977 \def\MT@check@rlist@#1#2#3{%
3978 \def\@tempb{#1}%
3979 \def\@tempc{#2}%
3980 \MT@if@false
3981 \MT@exp@cs\MT@map@tlist@
3982 {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3983 \MT@check@range
3984 }

```

`\MT@check@range` ... recurse through the list of existing ranges.

```
3985 \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.

```
3986 \def\MT@check@range@#1#2#3{%
3987   \MT@ifdim{#2}=\m@ne{%
3988     \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
3989     \MT@ifdim\@tempb={#1}\MT@iftrue\relax
3990   }{%
```

- Item in list is a simple size, new item is a range.

```
3991     \MT@ifdim\@tempb>{#1}\relax{%
3992     \MT@ifdim\@tempc>{#1}{%
3993       \MT@iftrue
3994       \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3995     }\relax
3996   }%
3997 }%
3998 }{%
3999 \MT@ifdim\@tempc=\m@ne{%
```

- Item in list is a range, new item is a simple size.

```
4000   \MT@ifdim\@tempb<{#2}{%
4001     \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
4002   }\relax
4003 }{%
```

- Both items are ranges.

```
4004   \MT@ifdim\@tempb<{#2}{%
4005     \MT@ifdim\@tempc>{#1}{%
4006       \MT@iftrue
4007       \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
4008     }\relax
4009   }\relax
4010 }%
4011 }%
4012 \ifMT@if@
4013   \MT@ifstreq{#3}%
4014   {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
4015   \relax{%
4016     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
4017       ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
4018       list ~#3' for font \@tempa,\MessageBreak size \@tempb}%
4019   }%
```

If we've already found a conflict with this item, we can skip the rest of the list.

```
4020   \expandafter\MT@tlist@break
4021   \fi
4022 }
```

## 14.4 Package options

### 14.4.1 Declaring the options

`\ifMT@opt@expansion` Keep track of whether the user explicitly set these options.

```
\ifMT@opt@auto 4023 \newif\ifMT@opt@expansion
\ifMT@opt@DVI
```

```

4024 \newif\ifMT@opt@auto
4025 \newif\ifMT@opt@DVI

\MT@optwarn@admissible    Some warnings.

4026 \def\MT@optwarn@admissible#1#2{%
4027   \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
4028     `#2'. Assuming `false'}%
4029 }

\MT@optwarn@nan

4030 </package>
4031 <*package|letterspace>
4032 <plain>\MT@requires@latex1{
4033 \def\MT@optwarn@nan#1#2{%
4034   \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
4035     Using default value of \number\@nameuse{MT@#2@default}}%
4036 }
4037 <plain>\relax
4038 </package|letterspace>
4039 <*package>

\MT@opt@def@set

4040 \def\MT@opt@def@set#1{%
4041   \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
4042     \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
4043   }{%
4044     \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
4045     \MT@warning@n1{The #1 set `~\MT@val' is undeclared.\MessageBreak
4046       Using set `~\@nameuse{MT@\@tempb @setname}' instead}%
4047   }%
4048 }

expansion and protrusion may be true, false, compatibility, nocompatibility
and/or a <set name>.

4049 \MT@map@clist@n{protrusion,expansion}{%
4050   \define@key{MT}{#1}[true]{%
4051     \csname MT@opt@#1true\endcsname
4052     \MT@map@clist@n{##1}{%
4053       \KV@esp@def\MT@val{###1}%
4054       \MT@ifempty\MT@val\relax{%
4055         \csname MT@#1true\endcsname
4056         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4057         \MT@ifstreq\MT@val{true}\relax
4058       }{%
4059         \MT@ifstreq\MT@val{false}{%
4060           \csname MT@#1false\endcsname
4061         }{%
4062           \MT@ifstreq\MT@val{compatibility}{%
4063             \MT@let@nc{MT@\@tempb @level}\@ne
4064           }{%
4065             \MT@ifstreq\MT@val{nocompatibility}{%
4066               \MT@let@nc{MT@\@tempb @level}\tw@
4067             }{%

```

If everything failed, it should be a set name.

```

4068       \MT@opt@def@set{#1}%
4069     }%
4070   }%
4071 }%
4072 }%
4073 }%
4074 }%
4075 }%
4076 }

```

activate is a shortcut for protrusion and expansion.

```
4077 \define@key{MT}{activate}[true]{%
4078   \setkeys{MT}{protrusion={#1}}%
4079   \setkeys{MT}{expansion={#1}}%
4080 }
```

spacing, kerning and tracking do not have a compatibility level.

```
4081 \MT@map@clist@n{spacing,kerning,tracking}{%
4082   \define@key{MT}{#1}[true]{%
4083     \MT@map@clist@n{##1}{%
4084       \KV@sp@def\MT@val{###1}%
4085       \MT@ifempty\MT@val\relax{%
4086         \csname MT@#1true\endcsname
4087         \MT@ifstreq\MT@val{true}\relax
4088         {%
4089           \MT@ifstreq\MT@val{false}{%
4090             \csname MT@#1false\endcsname
4091             }{%
4092               \edef\@tempb{\csname MT@rbba@#1\endcsname}%
4093               \MT@opt@def@set{#1}%
4094             }%
4095           }%
4096         }%
4097       }%
4098     }%
4099 }
```

`\MT@def@bool@opt` The true/false options: draft, final (may be inherited from the class options), auto, selected, babel, DVIoutput, defersetup, copyfonts.

```
4100 \def\MT@def@bool@opt#1#2{%
4101   \define@key{MT}{#1}[true]{%
4102     \def\@tempa{##1}%
4103     \MT@ifstreq\@tempa{true}\relax{%
4104       \MT@ifstreq\@tempa{false}\relax{%
4105         \MT@optwarn@admissible{##1}{#1}%
4106         \def\@tempa{false}%
4107       }%
4108     }%
4109     #2%
4110   }%
4111 }
```

Boolean options that only set the switch.

```
4112 \MT@map@clist@n{draft,selected,babel}{%
4113   \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
4114 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotru}
```

The DVIoutput option will change `\pdfoutput` immediately to minimise the risk of confusing other packages.

```
4115 </package>
4116 <*pdfTeX-def|luatex-def|xetex-def>
4117 <luatex-def>\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
4118 \MT@def@bool@opt{DVIoutput}{%
4119   \csname if\@tempa\endcsname
4120 <*pdfTeX-def|luatex-def>
4121   \ifnum\pdfoutput>\z@ \MT@opt@DVITrue \fi
4122   \pdfoutput\z@
4123   \else
4124     \ifnum\pdfoutput<\@ne \MT@opt@DVITrue \fi
4125     \pdfoutput\@ne
4126 </pdfTeX-def|luatex-def>
4127 <xetex-def> \MT@warning@n1{Ignoring `DVIoutput' option}%
4128   \fi
4129 }
```

```
4130 </pdftex-def|luatex-def|xetex-def>
```

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.

```
4131 <*package>
4132 \MT@def@bool@opt{defersetup}{%
4133   \csname if\@tempa\endcsname \else
4134     \AtEndOfPackage{%
4135       \MT@setup@
4136       \let\MT@setup@\empty
4137       \let\MT@addto@setup\@firstofone
4138     }%
4139   \fi
4140 }
4141 </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.

```
4142 <*pdftex-def|luatex-def>
4143 <pdftex-def>\MT@requires@pdftex7{
4144   \MT@def@bool@opt{copyfonts}{%
4145     \csname if\@tempa\endcsname
4146       \MT@gllet\MT@copy@font\MT@copy@font@
4147     \else
4148       \MT@gllet\MT@copy@font\relax
4149     \fi
4150   }
4151 <pdftex-def>}{
4152 </pdftex-def|luatex-def>
4153 <*pdftex-def|xetex-def>
4154   \MT@def@bool@opt{copyfonts}{%
4155     \csname if\@tempa\endcsname
4156       \MT@error
4157 <pdftex-def>      {The pdftex version you are using is too old\MessageBreak
4158 <pdftex-def>      to use the `copyfonts' option}{Upgrade pdftex.}%
4159 <xetex-def>      {The `copyfonts' option does not work with xetex}
4160 <xetex-def>      {Use pdftex or luatex instead.}%
4161     \fi
4162   }
4163 <pdftex-def>}{
4164 </pdftex-def|xetex-def>
```

`final` is the opposite to `draft`.

```
4165 <*package>
4166 \MT@def@bool@opt{final}{%
4167   \csname if\@tempa\endcsname
4168     \MT@draftfalse
4169   \else
4170     \MT@drafttrue
4171   \fi
4172 }
```

For verbose output, we redefine `\MT@vinfo`.

```
4173 \define@key{MT}{verbose}[true]{%
4174   \let\MT@vinfo\MT@info@n1
4175   \def\@tempa{#1}%
4176   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
4177 \MT@ifstreq\@tempa{errors}{%
4178 \let\MT@warning \MT@warn@err
4179 \let\MT@warning@n1\MT@warn@err
4180 }{%
4181 \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
4182 \MT@ifstreq\@tempa{silent}{%
4183 \let\MT@warning \MT@info
4184 \let\MT@warning@n1\MT@info@n1
4185 }{%
4186 \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
4187 }%
4188 }%
4189 }%
4190 }
4191 </package>
```

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```
4192 <*package|letterspace>
4193 <plain>\MT@requires@latex1{
4194 \MT@map@clist@n{%
4195 <package> stretch,shrink,step,%
4196 letterspace}{%
4197 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
4198 \def\@tempa{##1 }%
```

No nonsense in \MT@factor et al.? A space terminates the number.

```
4199 \MT@ifint\@tempa
4200 {\MT@edef@n{MT@#1}{\@tempa}}%
4201 {\MT@optwarn@nan{##1}{#1}}%
4202 }%
4203 }
4204 <plain>\relax
4205 </package|letterspace>
```

factor will define the protrusion factor only.

```
4206 <*package>
4207 \define@key{MT}{factor}[\MT@factor@default]{%
4208 \def\@tempa{#1 }%
4209 \MT@ifint\@tempa
4210 {\edef\MT@pr@factor{\@tempa}}
4211 {\MT@optwarn@nan{#1}{factor}}%
4212 }
```

Unit for protrusion codes.

```
4213 \define@key{MT}{unit}[character]{%
4214 \def\@tempa{#1}%
4215 \MT@ifstreq\@tempa{character}\relax{%
4216 \MT@ifdimen\@tempa
4217 {\let\MT@pr@unit\@tempa}%
4218 {\MT@warning@n1{`\@tempa' is not a dimension.\MessageBreak
4219 Ignoring it and setting values relative to\MessageBreak
4220 character widths}}%
4221 }%
4222 }
```

#### 14.4.2 Loading the definition file

\MT@endinput Abort if no capable engine found.

```
4223 \let\MT@endinput\relax
4224 \ifx\MT@engine\relax
```



```

4225 \MT@warning@nl{You don't seem to be using pdfTeX, luatex or xetex.\MessageBreak
4226   \MT@MT' only works with these engines.\MessageBreak
4227   I will quit now}
4228 \MT@clear@options
4229 \else

```

Otherwise load the engine-specific code (as strewn across this file).

```

4230 \input{microtype-\MT@engine tex.def}
4231 \fi
4232 \MT@endinput

```

### 14.4.3 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<sub>E</sub>X systems have switched to the pdfT<sub>E</sub>X engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfT<sub>E</sub>X.)

```

4233 \MT@protrusiontrue
4234 </package>
4235 <*pdfTeX-def|luatex-def>
4236 \ifnum\pdfoutput<\@ne \else

```

Also, we only enable expansion by default if pdfT<sub>E</sub>X can expand the fonts automatically.

```

4237 <pdfTeX-def> \MT@requires@pdfTeX4{
4238   \MT@expansiontrue
4239 <pdfTeX-def> \MT@autottrue
4240 <pdfTeX-def> } \relax
4241 \fi
4242 <luatex-def> \MT@autottrue
4243 </pdfTeX-def|luatex-def>

```

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  
 \MT@get@config to define a no-op for the regular option processing later.

```

4244 <*package>
4245 \define@key{MT}{config}[]{\relax}
4246 \def\MT@get@config#1config=#2,#3\@nil{%
4247   \MT@ifempty{#2}%
4248   {\def\MT@config@file{\MT@MT.cfg}}%
4249   {\def\MT@config@file{#2.cfg}}%
4250 }
4251 \expandafter\expandafter\expandafter\MT@get@config
4252 \csname opt@\@currname.\@current\endcsname,config=,\@nil

```

Load the file.

```

4253 \IfFileExists{\MT@config@file}{%
4254   \MT@info@nl{Loading configuration file \MT@config@file}%
4255   \MT@begin@catcodes
4256   \let\MT@begin@catcodes\relax
4257   \let\MT@end@catcodes\relax
4258   \let\MT@curr@file\MT@config@file
4259   \input{\MT@config@file}%
4260   \endgroup
4261 }{\MT@warning@nl{%
4262   Could not find configuration file \MT@config@file!\MessageBreak
4263   This will almost certainly cause undesired results.\MessageBreak
4264   Please fix your installation}%
4265 }

```

`\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).

```

4266 \def\MT@check@active@set#1{%
4267   \MT@ifdefined@n@TF{MT@#1@setname}{%
4268     \MT@info@n1{Using \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}'}%
4269   }{%
4270     \MT@ifdefined@n@TF{MT@default@#1@set}{%
4271       \MT@gl@et@nn{MT@#1@setname}{MT@default@#1@set}%
4272       \MT@info@n1{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}'}%
4273     }{%

```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '@', and issue a warning.

```

4274     \MT@gdef@n{MT@#1@setname}{@}%
4275     \MT@warning@n1{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
4276                   \MessageBreak Using empty set}%
4277   }%
4278 }%
4279 }

```

#### 14.4.4 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 provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

```

4280 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
4281   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
4282   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
4283 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

#### 14.4.5 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`, `\MT@define@optionX`

expansion, activate, tracking, spacing and kerning. Specifying font sets is not allowed.

```

4284 \def\microtypesetup{\setkeys{MT}}
4285 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
4286 

```

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.

```

4296   \MT@checksetup{#1}{%
4297     \@tempcnta=\csname MT@\@tempb @level\endcsname
4298     \MT@vinfo{Enabling #1
4299       (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
4300   }%
4301 }{%
4302   \MT@ifstreq\MT@val{false}{%
4303     \@tempcnta=\z@
4304     \MT@vinfo{Disabling #1\on@line}%
4305   }{%
4306     \MT@ifstreq\MT@val{compatibility}{%
4307       \MT@checksetup{#1}{%
4308         \@tempcnta=\@ne
4309         \MT@let@nc{MT@\@tempb @level}\@ne
4310         \MT@vinfo{Setting #1 to level 1\on@line}%
4311       }%
4312     }{%
4313       \MT@ifstreq\MT@val{nocompatibility}{%
4314         \MT@checksetup{#1}{%
4315           \@tempcnta=\tw@
4316           \MT@let@nc{MT@\@tempb @level}\tw@
4317           \MT@vinfo{Setting #1 to level 2\on@line}%
4318         }%
4319       }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
4320         {Use any of `true', `false', `compatibility' or
4321         `nocompatibility'.}%
4322     }%
4323   }%
4324 }%
4325 }%
4326 \ifnum\@tempcnta>\m@ne
4327   #2\@tempcnta\relax
4328 \fi
4329 }%
4330 }%
4331 }%
4332 }

```

\MT@checksetup Test whether the feature wasn't disabled in the package options.

```

4333 \def\MT@checksetup#1{%
4334   \csname ifMT@#1\endcsname
4335   \expandafter@firstofone
4336   \else
4337     \MT@error{You cannot enable #1 if it was disabled\MessageBreak
4338       in the package options}{Load microtype with #1 enabled.}%
4339     \expandafter@gobble

```

```

4340   \fi
4341 }

4342 \MT@define@optionX{protrusion}\MT@protrudechars
4343 </pdftex-def|luatex-def|xetex-def>
4344 <*pdftex-def|luatex-def>
4345 \MT@define@optionX{expansion}\MT@adjustspacing

\MT@protrudechars
\MT@adjustspacing 4346 <*luatex-def>
4347 \MT@requires@luatex4{
4348   \let\pdfprotrudechars\protrudechars
4349   \let\pdfadjustspacing\adjustspacing
4350 } \relax
4351 </luatex-def>
4352 \let\MT@protrudechars\pdfprotrudechars
4353 \let\MT@adjustspacing\pdfadjustspacing
4354 </pdftex-def|luatex-def>
4355 <*xetex-def>
4356 \let\MT@protrudechars\XeTeXprotrudechars
4357 \define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
4358 </xetex-def>

\MT@define@optionX{ The same for tracking, spacing and kerning, which do not have a compatibility
                    level.
4359 <*pdftex-def|luatex-def>
4360 <pdftex-def>\MT@requires@pdftex6{
4361 <luatex-def>\MT@requires@luatex3{
4362   \def\MT@define@optionX@#1#2{%
4363     \define@key{MTX}{#1}[true]{%
4364       \MT@map@clist@n{##1}{%
4365         \KV@sp@def\MT@val{###1}%
4366         \MT@ifempty\MT@val\relax{%
4367           \@tempcnta=\m@ne
4368           \MT@ifstreq\MT@val{true}{%
4369             \MT@checksetup{#1}{%
4370               \@tempcnta=\@ne
4371               \MT@vinfo{Enabling #1\on@line}%
4372             }%
4373           }%
4374           \MT@ifstreq\MT@val{false}{%
4375             \@tempcnta=\z@
4376             \MT@vinfo{Disabling #1\on@line}%
4377           }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
4378             {Use either `true' or `false'}}%
4379         }%
4380       }%
4381       \ifnum\@tempcnta>\m@ne
4382         #2\relax
4383       \fi
4384     }%
4385   }%
4386 }%
4387 }

```

We cannot simply let \MT@tracking relax, since this may select the already letter-spaced font instance.

```

4388 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
4389                               \else \let\MT@tracking\MT@tracking@ \fi}
4390 <pdftex-def> \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
4391 <pdftex-def> \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
4392 <pdftex-def> \pdfappendkern\@tempcnta}
4393 }{
4394 </pdftex-def|luatex-def>
4395 <*pdftex-def|luatex-def|xetex-def>

```

Disable for older pdfTeX versions and for X<sub>Y</sub>TeX and LuaTeX.

```

4396 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
4397 <luatex-def>
4398 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
4399 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
4400 <pdfTeX-def>
4401 \define@key{MTX}{activate}[true]{%
4402   \setkeys{MTX}{protrusion={#1}}%
4403   <pdfTeX-def|luatex-def> \setkeys{MTX}{expansion={#1}}%
4404 }
4405 </pdfTeX-def|luatex-def|xetex-def>

```

`\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. This is *undocumented*, as it completely deprives us of the possibility to act – we're blind and paralysed.

```

4406 <*package>
4407 \let\MT@saved@setupfont\MT@setupfont

4408 \define@key{MTX}{disable}[]{%
4409   \MT@info{Inactivate `~\MT@MT' package}%
4410   \let\MT@setupfont\relax
4411 }
4412 \define@key{MTX}{enable}[]{%
4413   \MT@info{Reactivate `~\MT@MT' package}%
4414   \let\MT@setupfont\MT@saved@setupfont
4415 }
4416 </package>

```

#### 14.4.6 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

4417 <*package|letterspace>
4418 <plain>\MT@requires@latex1{
4419 \def\MT@ProcessOptionsWithKV#1{%
4420   \let\MT@temp\relax
4421   \let\MT@temp@empty
4422 <plain> \MT@requires@latex2{
4423   \MT@map@clist@c@eclassoptionslist{%
4424     \def\CurrentOption{##1}%
4425     \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
4426       \edef\MT@temp{\MT@temp,\CurrentOption,}%
4427       \expandtwoargs\@removeelement\CurrentOption
4428       \unusedoptionlist\unusedoptionlist
4429     }%
4430   }%
4431   \edef\MT@temp{\noexpand\setkeys{#1}%
4432     {\MT@temp\@optionlist{\@currname.\@currentx}}}%

```

`eplain` can handle package options.

```

4433 <*plain>
4434 }{\edef\MT@temp{\noexpand\setkeys{#1}%
4435   {\csname usepkg@options@usepkg@pkg@endcsname}}}
4436 </plain>
4437 \MT@temp
4438 \MT@clear@options
4439 }

```

`\MT@getkey` For key=val in class options.

```

4440 \def\MT@getkey#1=#2\@nil{#1}
4441 \MT@ProcessOptionsWithKV{MT}
4442 <plain>\relax

```

```

4443 </package|letterspace>
4444 <*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).

```

4445 \MT@addto@setup{%
4446 \ifMT@draft

```

We disable most of what we've just defined in the 4446 lines above if we are running in draft mode.

```

4447 \MT@warning@nl{'draft' option active.\MessageBreak
4448           Disabling all micro-typographic extensions.\MessageBreak
4449           This might lead to different line and page breaks}%
4450 \let\MT@setupfont\relax
4451 \renewcommand*{\LoadMicrotypeFile[1]}{}%
4452 \renewcommand*\microtypesetup[1]{}%
4453 \renewcommand*\microtypecontext[1]{}%
4454 \renewcommand*\lsstyle{}%
4455 \else
4456 \MT@setup@PDF
4457 \MT@setup@copies

```

Fix the font sets.

```

4458 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
4459 \MT@setup@protrusion
4460 \MT@setup@expansion
4461 \MT@setup@tracking
4462 \MT@setup@warntracking
4463 \MT@setup@spacing
4464 \MT@setup@kerning
4465 \MT@setup@noligatures
4466 }
4467 </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!

```

4468 <*pdftex-def|luatex-def>
4469 \def\MT@setup@PDF{%
4470 \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
4471           \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
4472 }

```

\MT@setup@copies Working on font copies?

```

4473 \def\MT@setup@copies{%
4474 \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
4475 }
4476 </pdftex-def|luatex-def>
4477 <*xetex-def>
4478 \let\MT@setup@PDF\relax
4479 \let\MT@setup@copies\relax
4480 </xetex-def>

```

\MT@setup@protrusion Protrusion.

```

4481 <*pdftex-def|xetex-def|luatex-def>
4482 \def\MT@setup@protrusion{%
4483 \ifMT@protrusion

```

```

4484 \edef\MT@active@features{\MT@active@features,pr}%
4485 \MT@protrudechars\MT@pr@level
4486 \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)%
4487 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
4488 factor: \number\MT@pr@factor\fi
4489 \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
4490 \MT@check@active@set{pr}%
4491 \else
4492 \let\MT@protrusion\relax
4493 \MT@info@n1{No character protrusion}%
4494 \fi
4495 }
4496 </pdfTeX-def|xetex-def|luatex-def>

```

`\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.

```

4497 <(*pdfTeX-def|luatex-def)>
4498 \def\MT@setup@expansion{%
4499 \ifnum\pdfoutput<\@ne
4500 \ifMT@opt@expansion
4501 <(*luatex-def)>
4502 \ifMT@expansion
4503 \MT@requires@luatex3{%
4504 \MT@warning@n1{Font expansion doesn't work properly with luatex in\MessageBreak
4505 DVI mode: the glyphs won't be actually transformed,\MessageBreak
4506 but will only be shifted. You might want to use\MessageBreak
4507 pdfLaTeX instead. I'll continue anyway ..}%
4508 %\MT@expansionfalse
4509 }\relax
4510 \fi
4511 </luatex-def>
4512 \else
4513 \MT@expansionfalse
4514 \fi
4515 \fi
4516 \ifMT@expansion

```

Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.

```

4517 \ifnum\MT@stretch=\m@ne
4518 \let\MT@stretch\MT@stretch@default
4519 \fi

```

If shrink has not been specified, it will inherit the value from stretch.

```

4520 \ifnum\MT@shrink=\m@ne
4521 \let\MT@shrink\MT@stretch
4522 \fi

```

If step has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for microtype.pdf with step=1 compared to step=5). With older versions, we set it to min(stretch,shrink)/5, rounded off, minimum value 1.

```

4523 \ifnum\MT@step=\m@ne
4524 <pdfTeX-def> \MT@requires@pdfTeX6{%
4525 \def\MT@step{1 }%
4526 <(*pdfTeX-def)>
4527 }{%
4528 \ifnum\MT@stretch>\MT@shrink

```

```

4529     \ifnum\MT@shrink=\z@
4530     \@tempcnta=\MT@stretch
4531     \else
4532     \@tempcnta=\MT@shrink
4533     \fi
4534   \else
4535     \ifnum\MT@stretch=\z@
4536     \@tempcnta=\MT@shrink
4537     \else
4538     \@tempcnta=\MT@stretch
4539     \fi
4540   \fi
4541   \divide\@tempcnta 5\relax
4542   \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
4543   \edef\MT@step{\number\@tempcnta\space}%
4544 }%
4545 (/pdfTeX-def)
4546 \fi
4547 \ifnum\MT@step=\z@
4548   \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
4549     Setting it to one}%
4550   \def\MT@step{1}%
4551 \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.

```

4552   \let\MT@auto\empty
4553   \ifMT@auto

```

We turn off automatic expansion if output mode is DVI.

```

4554 (*pdfTeX-def)
4555   \MT@requires@pdfTeX4{%
4556     \ifnum\pdfoutput<\@ne
4557     \ifMT@opt@auto
4558       \MT@error{%
4559         Automatic font expansion only works for PDF output.\MessageBreak
4560         However, you are creating a DVI file}
4561       {If you have created expanded fonts instances, remove `auto' from%
4562         \MessageBreak the package options. Otherwise, you have to switch
4563         off expansion\MessageBreak completely.}%
4564     \fi
4565     \MT@autofalse
4566   \else
4567     \def\MT@auto{autoexpand}%
4568   \fi

```

Also, if pdfTeX is too old.

```

4569   }{%
4570     \MT@error{%
4571       The pdfTeX version you are using is too old for\MessageBreak
4572       automatic font expansion}%
4573     {If you have created expanded fonts instances, remove `auto' from\MessageBreak
4574       the package options. Otherwise, you have to switch off expansion\MessageBreak
4575       completely, or upgrade pdfTeX to version 1.20 or newer.}%
4576     \MT@autofalse
4577     \def\MT@auto{1000}%
4578   }%
4579 (/pdfTeX-def)
4580 (luaTeX-def)   \MT@requires@luaTeX3\relax{\def\MT@auto{autoexpand}}%
4581 \else

```



4582 *<pdf<sub>tex</sub>-def>*

No automatic expansion.

```

4583     \MT@requires@pdftex4\relax{%
4584     \def\MT@auto{1000 }%
4585     }%
4586 </pdftex-def>
4587 <luatex-def>
4588     \MT@requires@luatex3{%
4589     \ifMT@opt@auto
4590     \MT@error{Non-automatic font expansion does not work with\MessageBreak
4591             luatex}{Remove `auto=false' from the package options, or use pdftex.}%
4592     \MT@autotue
4593     \fi
4594     }\relax
4595 </luatex-def>
4596     \fi

```

Choose the appropriate macro for selected expansion.

```

4597     \ifMT@selected
4598     \let\MT@set@ex@codes\MT@set@ex@codes@s
4599     \else
4600     \let\MT@set@ex@codes\MT@set@ex@codes@n
4601     \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdf<sub>TeX</sub> error.

```

4602     \ifnum\MT@stretch=\z@
4603     \ifnum\MT@shrink=\z@
4604     \MT@warning@n{%
4605         Both the stretch and shrink limit are set to zero.\MessageBreak
4606         Disabling font expansion}%
4607     \MT@expansionfalse
4608     \fi
4609     \fi
4610     \fi
4611     \ifMT@expansion
4612     \edef\MT@active@features{\MT@active@features,ex}%
4613     \MT@adjustspacing\MT@ex@level
4614     \MT@info@n{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
4615             (level \number\MT@ex@level),\MessageBreak
4616             stretch: \number\MT@stretch, shrink: \number\MT@shrink,
4617             step: \number\MT@step, \ifMT@selected\else non-\fi selected}%

```

\MT@check@step      Check whether stretch and shrink are multiples of step.

```

4618     \def\MT@check@step##1{%
4619     \@tempcnta=\csname MT@##1\endcsname
4620     \divide\@tempcnta \MT@step
4621     \multiply\@tempcnta \MT@step
4622     \ifnum\@tempcnta=\csname MT@##1\endcsname\else
4623     \MT@warning@n{The ##1 amount is not a multiple of step.\MessageBreak
4624             The effective maximum ##1 is \the\@tempcnta\space
4625             (step \number\MT@step)}%
4626     \fi
4627     }%
4628     \MT@check@step{stretch}%
4629     \MT@check@step{shrink}%
4630     \MT@check@active@set{ex}%

```

\showhyphens      Inside \showhyphens, font expansion should be disabled. (Since 2017/01/10, the L<sub>A</sub>T<sub>E</sub>X format contains a different version for X<sub>Y</sub>T<sub>E</sub>X, but since expansion doesn't work with X<sub>Y</sub>T<sub>E</sub>X, we don't have to bother.) Since 2019/10/01, the command is robust.

```

4631     \MT@ifdefined@n@TF{showhyphens }{%
4632     \def\MT@temp##1##2{%

```

```

4633     \expandafter\CheckCommand\csname showhyphens \endcsname[1]{##1}%
4634     \DeclareRobustCommand\showhyphens[1]{##2}}%
4635   }{%
4636     \def\MT@temp##1##2{%
4637       \CheckCommand*\showhyphens[1]{##1}%
4638       \gdef\showhyphens###1{##2}}%
4639   }%
4640   \MT@temp
4641   {\setbox0\vbox{\color@begingroup
4642     \everypar{}\parfillskip\z@skip
4643     \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4644     \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}
4645   {\setbox0\vbox{\color@begingroup\pdfadjustspacing\z@
4646     \everypar{}\parfillskip\z@skip
4647     \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4648     \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%
4649   \else
4650     \let\MT@expansion\relax
4651     \MT@info@nl{No font expansion}%
4652   \fi
4653 }
4654 </pdfTeX-def|LaTeX-def>
4655 <*XeTeX-def>
4656 \def\MT@setup@expansion{%
4657   \ifMT@expansion
4658     \ifMT@opt@expansion
4659       \MT@error{Font expansion does not work with XeTeX}%
4660       {Use pdfTeX or LaTeX instead.}%
4661   \fi
4662   \fi
4663 }
4664 </XeTeX-def>

```

\MT@setup@tracking      Tracking, spacing and kerning.

```

4665 <*pdfTeX-def|LaTeX-def>
4666 <pdfTeX-def>\MT@requires@pdfTeX6{%
4667 <LaTeX-def>\MT@requires@LaTeX3{%
4668   \def\MT@setup@tracking{%
4669     \ifMT@tracking
4670       \edef\MT@active@features{\MT@active@features,tr}%
4671       \MT@info@nl{Tracking enabled}%
4672       \MT@check@active@set{tr}%

```

Enable protrusion for compensation at the line edges.

```

4673     \ifMT@protrusion\else\MT@protrudechars\@ne\fi
4674   \else
4675     \let\MT@tracking\relax
4676     \MT@info@nl{No adjustment of tracking}%
4677   \fi
4678 }
4679 </pdfTeX-def|LaTeX-def>

```

\MT@setup@spacing

```

4680 <*pdfTeX-def>
4681 \def\MT@setup@spacing{%
4682   \ifMT@spacing
4683     \edef\MT@active@features{\MT@active@features,sp}%
4684     \pdfadjustinterwordglue\@ne
4685     \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.

```

4686   \MT@with@package@T{ragged2e}{%
4687     \MT@warning@nl{You are using the 'ragged2e' package. \MessageBreak

```

```

4688         Adjustment of interword spacing may lead to\MessageBreak
4689         undesired results when used with `ragged2e'.\MessageBreak
4690         In this case, disable the `spacing' option}%
4691     }%
4692     \MT@check@active@set{sp}%
4693 \else
4694     \let\MT@spacing\relax
4695     \MT@info@nl{No adjustment of interword spacing}%
4696 \fi
4697 }

\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.16
4698 \def\MT@setup@spacing@check{%
4699     \ifMT@spacing
4700     \ifMT@babel \else
4701         \ifnum\sfcode`. > 1500
4702             \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
4703                 \MT@warning@nl{%
4704                     \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
4705                     interword spacing will disable it. You might want\MessageBreak
4706                     to add `\'@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
4707                     to your preamble}%
4708                 }%
4709             \fi
4710         \fi
4711     \fi
4712 }

\MT@setup@kerning
4713 \def\MT@setup@kerning{%
4714     \ifMT@kerning
4715         \edef\MT@active@features{\MT@active@features,kn}%
4716         \pdfprependkern\@ne
4717         \pdfappendkern\@ne
4718         \MT@info@nl{Adjustment of character kerning enabled}%
4719         \MT@check@active@set{kn}%
4720     \else
4721         \let\MT@kerning\relax
4722         \MT@info@nl{No adjustment of character kerning}%
4723     \fi
4724 }
4725 </pdfTeX-def>

\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.
4726 <pdfTeX-def|luatex-def>}{
4727 <*luatex-def>
4728     \def\MT@setup@tracking{%
4729         \ifMT@tracking
4730             \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
4731                 or newer. Switching it off}{Upgrade luatex.}%
4732             \MT@trackingfalse
4733             \MT@let@nc{MT@tracking}\relax
4734         \else
4735             \MT@info@nl{No adjustment of tracking (luatex too old)}%
4736         \fi
4737     }
4738 }
4739 </luatex-def>

```

16 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

```

4740 <pdfTex-def|xetex-def|luatex-def>
4741 \def\MT@error@doesnt@work#1{%
4742   \csname ifMT@#1\endcsname
4743   \MT@error{The #1 feature only works with pdfTeX 1.40\MessageBreak
4744     or newer. Switching it off}
4745   <pdfTex-def> {Upgrade pdfTeX.}%
4746   <luatex-def|xetex-def> {Use pdfTeX instead.}%
4747   \csname MT@#1false\endcsname
4748   \MT@let\nc{MT@#1}\relax
4749   \else
4750     \MT@info\nc{No adjustment of #1%
4751   <pdfTex-def> \space(pdfTeX too old)%
4752   }%
4753   \fi
4754 }
4755 <pdfTex-def|xetex-def> \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
4756 \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
4757 \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
4758 <pdfTex-def>
4759 </pdfTex-def|xetex-def|luatex-def>

```

\MT@setup@warntracking

```

4760 <letterspace>\MT@addto@setup
4761 <pdfTex-def|luatex-def>\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.

```

4762 <pdfTex-def|luatex-def|letterspace>
4763 {%
4764 <pdfTex-def|letterspace>
4765 \ifnum\pdfoutput<\@ne
4766 \def\MT@warn@tracking@DVI{%
4767 <letterspace> \MT@pdf@or@lua{%
4768 \MT@warning\nc{
4769   You are using tracking/letterspacing in DVI mode.\MessageBreak
4770   This will probably not work, unless the post-\MessageBreak
4771   processing program (dvips, dvipdfm(x), ...) is\MessageBreak
4772   able to create the virtual fonts on the fly}%
4773 <letterspace> }\relax
4774 \MT@gl\let\MT@warn@tracking@DVI\relax
4775 }%
4776 \else
4777 </pdfTex-def|letterspace>
4778 \def\MT@warn@tracking@DVI{%
4779 \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
4780 \MT@gl\let\MT@warn@tracking@DVI\relax
4781 }%
4782 <pdfTex-def|letterspace> \fi
4783 \ifnum\MT@letterspace=\m@ne
4784 \let\MT@letterspace\MT@letterspace@default
4785 \else
4786 \MT@ls@too@large\MT@letterspace
4787 \fi
4788 }
4789 </pdfTex-def|luatex-def|letterspace>
4790 <xetex-def>\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.

```

4791 <pdfTex-def|luatex-def>
4792 \def\MT@setup@noligatures{%
4793 <pdfTex-def> \MT@requires@pdfTeX5{%
4794 \ifMT@noligatures \else

```

```

4795 \let\MT@noligatures\relax
4796 \fi
4797 <pdfTeX-def> }\relax
4798 }
4799 </pdfTeX-def|luatex-def>
4800 <xetex-def>\let\MT@setup@noligatures\relax

```

Remove the leading comma in \MT@active@features, and set the document switch to true.

```

4801 <*package>
4802 \MT@addto@setup{%
4803 \ifx\MT@active@features\empty \else
4804 \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
4805 \fi
4806 \MT@documenttrue
4807 }

```

\MT@set@babel@context Interaction with babel.

```

4808 \def\MT@set@babel@context#1{%
4809 \MT@ifdefined@n@TF{MT@babel@#1}{%
4810 \MT@info{*** Changing to language context `#1'\MessageBreak\on@line}%
4811 \expandafter\MT@exp@one@n\expandafter\microtypecontext
4812 \csname MT@babel@#1\endcsname
4813 }{%
4814 \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
4815 }%
4816 }

```

\MT@shorthandoff Active characters can only be switched off if babel isn't loaded after microtype.

```

4817 \ifpackageloaded{babel}{
4818 \def\MT@shorthandoff#1#2{%
4819 \MT@info@n1{Switching off #1 babel's active characters (#2)}%
4820 \shorthandoff{#2}}
4821 }{
4822 \def\MT@shorthandoff#1#2{%
4823 \MT@error{You must load `babel' before `~\MT@MT'}
4824 {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
4825 active characters.}}
4826 }

```

We patch the language switching commands to enable language-dependent setup.

```

4827 \MT@addto@setup{%
4828 \ifMT@babel
4829 \ifpackageloaded{babel}{%
4830 \MT@info@n1{Redefining babel's language switching commands}%
4831 \let\MT@orig@select@language\select@language
4832 \def\select@language#1{%
4833 \MT@orig@select@language{#1}%
4834 \MT@set@babel@context{#1}%
4835 }%
4836 \let\MT@orig@foreign@language\foreign@language
4837 \def\foreign@language#1{%
4838 \MT@orig@foreign@language{#1}%
4839 \MT@set@babel@context{#1}%
4840 }%
4841 \ifMT@kerning

```

Disable French babel's active characters.

```

4842 \MT@if@false
4843 \MT@with@babel@and@T{french} \MT@if@true
4844 \MT@with@babel@and@T{frenchb} \MT@if@true
4845 \MT@with@babel@and@T{français}\MT@if@true
4846 \MT@with@babel@and@T{canadien}\MT@if@true
4847 \MT@with@babel@and@T{acadian} \MT@if@true
4848 \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```
4849 \MT@if@false
4850 \MT@with@babel@and@T{turkish} \MT@if@true
4851 \ifMT@if@MT@shorthandoff{Turkish}{:=}\fi
4852 \fi
```

In case babel was loaded before microtype:

```
4853 \MT@set@babel@context\language
4854 }{%
4855 \MT@warning@n{You did not load the babel package.\MessageBreak
4856 The `babel' option won't have any effect}%
4857 }%
4858 \fi
4859 }
```

Now we close the \fi from \ifMT@draft.

```
4860 \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.

```
4861 \selectfont}
```

\MT@curr@file This is the current file (hopefully with the correct extension).

```
4862 \edef\MT@curr@file{\jobname.tex}
4863 </package>
```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```
4864 <*package|letterspace>
4865 <plain>\MT@requires@latex1{
4866 \AtBeginDocument{\MT@setup@ \MT@gl@et\MT@setup@ \@empty}
4867 <plain>}\relax
4868 </package|letterspace>
```

Must come at the very, very end.

```
4869 <package>\MT@ifdefined@c@T\MT@setup@spacing@check
4870 <package> {\AtBeginDocument{\MT@setup@spacing@check}}
```

Restore catcodes.

```
4871 <package|letterspace>\MT@restore@catcodes
```

That was that.

## 15 Configuration files

Let's now write the font configuration files.

```
4872 (*config)
4873
```

### 15.1 Font sets

We first declare some sets in the main configuration file.

```
4874 (*m-t)
4875 %%% -----
4876 %%% FONT SETS
4877
4878 \DeclareMicrotypeSet{all}
4879 { }
4880
4881 \DeclareMicrotypeSet{allmath}
4882 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }
4883
4884 \DeclareMicrotypeSet{alltext}
4885 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,TS1,EU1,EU2,TU} }
4886
4887 \DeclareMicrotypeSet{allmath-nott}
4888 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
4889   family   = {rm*,sf*}
4890 }
4891
4892 \DeclareMicrotypeSet{alltext-nott}
4893 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,TS1,EU1,EU2,TU},
4894   family   = {rm*,sf*}
4895 }
4896
4897 \DeclareMicrotypeSet{basicmath}
4898 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
4899   family   = {rm*,sf*},
4900   series    = {md*},
4901   size      = {normalsize,footnotesize,small,large}
4902 }
4903
4904 \DeclareMicrotypeSet{basictext}
4905 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,EU1,EU2,TU},
4906   family   = {rm*,sf*},
4907   series    = {md*},
4908   size      = {normalsize,footnotesize,small,large}
4909 }
4910
4911 \DeclareMicrotypeSet{smallcaps}
4912 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,TS1,EU1,EU2,TU},
4913   shape     = {sc*,si,scit}
4914 }
4915
4916 \DeclareMicrotypeSet{footnotesize}
4917 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,TS1,EU1,EU2,TU},
4918   size      = {-small}
4919 }
4920
4921 \DeclareMicrotypeSet{scriptsize}
4922 { encoding = {OT1,T1,T2A,Ly1,OT4,QX,T5,TS1,EU1,EU2,TU},
```

```

4923     size      = {-footnotesize}
4924   }
4925
4926 \DeclareMicrotypeSet{normal font}
4927   { font = */*/*/*/* }
4928

```

The default sets.

```

4929 %%% -----
4930 %%% DEFAULT SETS
4931
4932 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4933 \DeclareMicrotypeSetDefault[expansion]{alltext-nott}
4934 \DeclareMicrotypeSetDefault[spacing]{alltext-nott}
4935 \DeclareMicrotypeSetDefault[kerning]{alltext}
4936 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4937

```

## 15.2 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.

```

4938 %%% -----
4939 %%% FONT VARIANTS AND ALIASES
4940
4941 \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 Xe<sub>La</sub>TeX and Lua<sub>La</sub>TeX in the L<sup>A</sup>T<sub>E</sub>X format is TU, even if fontspec is not loaded.

```

4942
4943 \MT@if@false
4944 \ifx\UnicodeEncodingName\undefined\else
4945   \MT@if@fstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
4946 \fi
4947 \ifMT@fontspec\MT@if@true\fi
4948 \ifMT@if@
4949 % -- Computer/Latin Modern Roman
4950 \DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
4951   \else
4952 \DeclareMicrotypeAlias{lmr}{cmr}          % lmodern
4953 \fi

```

The Latin Modern fonts, the virtual fonts from the ae and zefonts and the eco and hfoldsty packages (oldstyle numerals), as well as lmodern, 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.

```

4954 \DeclareMicrotypeAlias{lmsy}{cmsy}      % "
4955 \DeclareMicrotypeAlias{lmm}{cmm}        % "
4956 \DeclareMicrotypeAlias{aer}{cmr}        % ae
4957 \DeclareMicrotypeAlias{zer}{cmr}        % zefonts

```



```

4958 \DeclareMicrotypeAlias{cmor}{cmr}          % eco
4959 \DeclareMicrotypeAlias{hfor}{cmr}           % hfoldsty
4960 \DeclareMicrotypeAlias{mlmr}{cmr}           % mlmodern
4961 \DeclareMicrotypeAlias{mlmsy}{cmsy}         % "
4962 \DeclareMicrotypeAlias{mlmm}{cmm}           % "

```

Another, new Computer Modern extension. The newcomputermodern package loads it by file name.

```

4963 \DeclareMicrotypeAlias{New Computer Modern}{Latin Modern Roman}
4964 \DeclareMicrotypeAlias{NewCM10-Book.otf}{Latin Modern Roman}
4965 \DeclareMicrotypeAlias{NewCM10-Regular.otf}{Latin Modern Roman}

```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the T<sub>E</sub>X Gyre fonts Pagella and Termes (formerly: qfonts).

```

4966 %% -- Palatino
4967 \DeclareMicrotypeAlias{pxr}{ppl}            % pxfonts
4968 \DeclareMicrotypeAlias{qpl}{ppl}            % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

```

4969 \DeclareMicrotypeAlias{fp9x}{pplx}         % FPL Neu
4970 \DeclareMicrotypeAlias{fp9j}{pplj}         % "

```

The newpx package, a replacement for pxfonts.

```

4971 \DeclareMicrotypeAlias{zpllf}{pplx}        % newpxtext
4972 \DeclareMicrotypeAlias{zplosf}{pplj}       % "
4973 \DeclareMicrotypeAlias{zpltlf}{pplx}       % "
4974 \DeclareMicrotypeAlias{zpltosf}{pplj}      % "

```

The domitian package.

```

4975 \DeclareMicrotypeAlias{Domitian-TLF}{pplx}% domitian
4976 \DeclareMicrotypeAlias{Domitian-T0sF}{pplj}% "

```

The OpenType versions:

```

4977 \DeclareMicrotypeAlias{Palatino Linotype}{Palatino}
4978 \DeclareMicrotypeAlias{Palatino LT Std}{Palatino}
4979 \DeclareMicrotypeAlias{TeX Gyre Pagella}{Palatino}
4980 \DeclareMicrotypeAlias{Domitian}{Palatino}
4981 \DeclareMicrotypeAlias{Asana Math}{Palatino}
4982 %% -- Times New Roman
4983 \DeclareMicrotypeAlias{txr}{ptm}            % txfonts

```

The newtx package, a replacement for txfonts.

```

4984 \DeclareMicrotypeAlias{ntxlf}{ptmx}        % newtxtext
4985 \DeclareMicrotypeAlias{ntxosf}{ptmj}       % "
4986 \DeclareMicrotypeAlias{ntxtlf}{ptmx}      % "
4987 \DeclareMicrotypeAlias{ntxtosf}{ptmj}     % "

```

The tempora package.

```

4988 \DeclareMicrotypeAlias{Tempora-TLF}{ptmx} % tempora
4989 \DeclareMicrotypeAlias{Tempora-T0sF}{ptmj} % "
4990 \DeclareMicrotypeAlias{qtm}{ptm}          % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

The step package.

```

4991 \DeclareMicrotypeAlias{STEP-TLF}{ptmx}    % step
4992 \DeclareMicrotypeAlias{STEP-T0sF}{ptmj}   % "

```

The stix and stix2 packages (the latter has departed a bit from being a Times clone, but still seems close enough).

```

4993 \DeclareMicrotypeAlias{stix}{ptm}         % stix
4994 \DeclareMicrotypeAlias{stix2}{ptm}        % stix2

```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (Times-NewRomanMT, TimesNRSevenMT), mtm (TimesSmallTextMT); pte (TimesEuropa); ptt (TimesTen); TimesEighteen; TimesModernEF.

MicroPress's Charter version (chmath).

```
4995 %% -- Charter
4996 \DeclareMicrotypeAlias{chr}{bch} % CH Math
```

The XCharter package extends the Charter fonts.

```
4997 \DeclareMicrotypeAlias{XCharter-TLF}{bch} % XCharter
4998 \DeclareMicrotypeAlias{XCharter-T0sF}{bch} % "
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
4999 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
5000 %% -- Garamond
5001 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

The garamondx package, an extension of URW Garamond, providing small caps and oldstyle figures.

```
5002 \DeclareMicrotypeAlias{zgmX}{ugm} % garamondx
5003 \DeclareMicrotypeAlias{zgmj}{ugm} % "
5004 \DeclareMicrotypeAlias{zgmI}{ugm} % "
5005 \DeclareMicrotypeAlias{zgmQ}{ugm} % "
```

Because a configuration file for Adobe Garamond wouldn't be permitted for T<sub>E</sub>X Live distribution, we use EB Garamond as the base font.

```
5006 \DeclareMicrotypeAlias{pad}{EBGaramond-LF}% Adobe Garamond
5007 \DeclareMicrotypeAlias{padx}{EBGaramond-TLF}% "
5008 \DeclareMicrotypeAlias{padj}{EBGaramond-T0sF}% "
5009 %% --
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
5010 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

The eulervm package virtually extends the Euler fonts.

```
5011 \DeclareMicrotypeAlias{zeur}{eur} % Euler VM
5012 \DeclareMicrotypeAlias{zeus}{eus} % "
```

Euro symbol fonts, to save some files.

```
5013 \DeclareMicrotypeAlias{zpeus}{zpeu} % Adobe Euro sans -> serif
5014 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
5015
```

## 15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
5016 %%% -----
5017 %%% INTERACTION WITH THE `babel' PACKAGE
5018
5019 \DeclareMicrotypeBabelHook
5020 {english,UKenglish,british,USenglish,american}
5021 {kerning=, spacing=nonfrench}
5022
5023 \DeclareMicrotypeBabelHook
5024 {french,français,acadian,canadien}
5025 {kerning=french, spacing=}
5026
5027 \DeclareMicrotypeBabelHook
5028 {turkish}
5029 {kerning=turkish, spacing=}
5030
```

## 15.4 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  $\text{\LaTeX}$  way, that is, when they have been assigned a slot in the font encoding with  $\text{\DeclareTextSymbol}$  or  $\text{\DeclareTextComposite}$ . Characters defined via  $\text{\chardef}$  are also possible.

Ligatures and  $\text{\mathchardef}$ 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  $\text{\XeTeX}$  or  $\text{\LuaTeX}$ , in contrast, it is advisable to use the proper Unicode characters, or the font-specific glyph names prefixed with '/' (cf. section 16).

## 15.5 Character inheritance

First the lists of inheriting characters. We only declare those characters that are the same on *both* sides, i.e., not Œ for O.

```
5031 </m-t>
5032 <*m-t|zpeu|mys>
5033 %%% -----
5034 %%% CHARACTER INHERITANCE
5035
5036 </m-t|zpeu|mys>
5037 <*m-t>
```

### 15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 ('fi' ligature), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
5038 \DeclareCharacterInheritance
5039 { encoding = OT1 }
5040 { f = {011}, % ff
5041   i = {\i},
5042   j = {\j},
5043   0 = {\0},
5044   o = {\o}
5045 }
5046
```

**15.5.2 T1**

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since L<sup>A</sup>T<sub>E</sub>X 2005/12/01 accessible as \IJ), 188 (‘ij’, \ij), Æ, æ, Œ, œ.

```

5047 \DeclareCharacterInheritance
5048 { encoding = T1 }
5049 { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
5050   a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
5051   C = {\`C,\'C,\c C,\v C},
5052   c = {\`c,\'c,\c c,\v c},
5053   D = {\v D,\DH},
5054   d = {\v d,\dj},
5055   E = {\`E,\'E,\^E,\"E,\k E,\v E},
5056   e = {\`e,\'e,\^e,\"e,\k e,\v e},
5057   f = {027}, % ff
5058   G = {\u G},
5059   g = {\u g},
5060   I = {\`I,\'I,\^I,\"I,\.I},
5061   i = {\`i,\'i,\^i,\"i,\i},
5062   j = {\j},
5063   L = {\L,\'L,\v L},
5064   l = {\l,\'l,\v l},
5065   N = {\`N,\~N,\v N},
5066   n = {\`n,\~n,\v n},
5067   O = {\0,\`0,\'0,\^0,\~0,\"0,\H 0},
5068   o = {\0,\`o,\'o,\^o,\~o,\"o,\H o},
5069   R = {\`R,\v R},
5070   r = {\`r,\v r},
5071   S = {\`S,\c S,\v S,\SS},
5072   s = {\`s,\c s,\v s},
5073   T = {\c T,\v T},
5074   t = {\c t,\v t},
5075   U = {\`U,\'U,\^U,\"U,\H U,\r U},
5076   u = {\`u,\'u,\^u,\"u,\H u,\r u},
5077   Y = {\`Y,\"Y},
5078   y = {\`y,\"y},
5079   Z = {\`Z,\.Z,\v Z},
5080   z = {\`z,\.z,\v z}

```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```

5081 %   - = {127},
5082   }
5083

```

**15.5.3 LY1**

More characters: 008 (‘fl’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```

5084 \DeclareCharacterInheritance
5085 { encoding = LY1 }
5086 { A = {\`A,\'A,\^A,\~A,\"A,\r A},
5087   a = {\`a,\'a,\^a,\~a,\"a,\r a},
5088   C = {\c C},
5089   c = {\c c},
5090   D = {\DH},
5091   E = {\`E,\'E,\^E,\"E},
5092   e = {\`e,\'e,\^e,\"e},
5093   f = {011}, % ff
5094   I = {\`I,\'I,\^I,\"I},
5095   i = {\`i,\'i,\^i,\"i,\i},
5096   L = {\L},
5097   l = {\l},
5098   N = {\~N},

```

```

5099     n = {\-n},
5100     O = {\^0,\^0,\^0,\^0,\^0,\^0},
5101     o = {\^o,\^o,\^o,\^o,\^o,\^o},
5102     S = {\v S},
5103     s = {\v s},
5104     U = {\^U,\^U,\^U,\^U},
5105     u = {\^u,\^u,\^u,\^u},
5106     Y = {\^Y,\^Y},
5107     y = {\^y,\^y},
5108     Z = {\v Z},
5109     z = {\v z}
5110 }
5111

```

#### 15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

5112 \DeclareCharacterInheritance
5113 { encoding = OT4 }
5114 { A = {\k A},
5115   a = {\k a},
5116   C = {\^C},
5117   c = {\^c},
5118   E = {\k E},
5119   e = {\k e},
5120   f = {011}, % ff
5121   i = {\i},
5122   j = {\j},
5123   L = {\L},
5124   l = {\l},
5125   N = {\^N},
5126   n = {\^n},
5127   O = {\0,\^0},
5128   o = {\o,\^o},
5129   S = {\^S},
5130   s = {\^s},
5131   Z = {\^Z,\^Z},
5132   z = {\^z,\^z},
5133   \textquotedblleft = "FF
5134 }
5135

```

#### 15.5.5 QX

The Central European QX encoding.<sup>17</sup> Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

5136 \DeclareCharacterInheritance
5137 { encoding = QX }
5138 { A = {\^A,\^A,\^A,\^A,\^A,\^A,\^A,\^A},
5139   a = {\^a,\^a,\^a,\^a,\^a,\^a,\^a,\^a},
5140   C = {\^C,\^C},
5141   c = {\^c,\^c},
5142   D = {\DH},
5143   E = {\^E,\^E,\^E,\^E,\^E,\^E},
5144   e = {\^e,\^e,\^e,\^e,\^e,\^e},
5145   f = {011}, % ff
5146   I = {\^I,\^I,\^I,\^I,\^I,\^I},
5147   i = {\^i,\^i,\^i,\^i,\^i,\^i},
5148   j = {\j},

```

17 Contributed by Maciej Eder.

```

5149 L = {\L},
5150 l = {\l},
5151 N = {\'N,\~N},
5152 n = {\'n,\~n},
5153 O = {\0,\`0,\'0,\^0,\-0,\"0},
5154 o = {\0,\`o,\'o,\^o,\-o,\"o},

```

The Romanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously<sup>18</sup>) been included in QX encoding. They are still kept for backwards compatibility.

```

5155 S = {\'S,\c S,\textcommabelow S,\v S},
5156 s = {\'s,\c s,\textcommabelow s,\v s},
5157 T = {\c T,\textcommabelow T},
5158 t = {\c t,\textcommabelow t},
5159 U = {\`U,\'U,\^U,\"U,\k U},
5160 u = {\`u,\'u,\^u,\"u,\k u},
5161 Y = {\'Y,\"Y},
5162 y = {\'y,\"y},
5163 Z = {\'Z,\Z,\v Z},
5164 z = {\'z,\Z,\v z},
5165 . = \textellipsis
5166 }
5167

```

### 15.5.6 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented characters that there is no room for any ligatures.

```

5168 \DeclareCharacterInheritance
5169 { encoding = T5 }
5170 { A = {\`A,\'A,\-A,\h A,\d A,\^A,\u A,
5171       \`Acircumflex,\'Acircumflex,\-Acircumflex,\hAcircumflex,\dAcircumflex,
5172       \`Abreve,\'Abreve,\-Abreve,\hAbreve,\dAbreve},
5173   a = {\`a,\'a,\-a,\h a,\d a,\^a,\u a,
5174       \`acircumflex,\'acircumflex,\-acircumflex,\hacircumflex,\dacircumflex,
5175       \`abreve,\'abreve,\-abreve,\habreve,\dabreve},
5176   D = {\DJ},
5177   d = {\dj},
5178   E = {\`E,\'E,\-E,\h E,\d E,\^E,
5179       \`Ecircumflex,\'Ecircumflex,\-Ecircumflex,\hEcircumflex,\dEcircumflex},
5180   e = {\`e,\'e,\-e,\h e,\d e,\^e,
5181       \`ecircumflex,\'ecircumflex,\-ecircumflex,\hecircumflex,\decircumflex},
5182   I = {\`I,\'I,\-I,\h I,\d I},
5183   i = {\`i,\'i,\-i,\h i,\d i,\i},
5184   O = {\`O,\'O,\-O,\h O,\d O,\^O,\horn O,
5185       \`Ocircumflex,\'Ocircumflex,\-Ocircumflex,\hOcircumflex,\dOcircumflex,
5186       \`Ohorn,\'Ohorn,\-Ohorn,\hOhorn,\dOhorn},
5187   o = {\`o,\'o,\-o,\h o,\d o,\^o,\horn o,
5188       \`ocircumflex,\'ocircumflex,\-ocircumflex,\hocircumflex,\docircumflex,
5189       \`ohorn,\'ohorn,\-ohorn,\hohorn,\dohorn},
5190   U = {\`U,\'U,\-U,\h U,\d U,\horn U,
5191       \`Uhorn,\'Uhorn,\-Uhorn,\hUhorn,\dUhorn},
5192   u = {\`u,\'u,\-u,\h u,\d u,\horn u,
5193       \`uhorn,\'uhorn,\-uhorn,\huhorn,\duhorn},
5194   Y = {\`Y,\'Y,\-Y,\h Y,\d Y},
5195   y = {\`y,\'y,\-y,\h y,\d y}
5196 }
5197

```

18 Cf. <http://tug.org/pipermail/tex-live/2008-August/017204.html>

### 15.5.7 EU1, EU2, TU

The EU1 (X<sub>Y</sub>TeX), EU2 (LuaTeX), and, since fontspec version 2.5, TU encodings are not well-defined in the sense that they don't contain a fixed number of glyphs, all of which must be present. OpenType fonts may contain thousands of glyphs, but we only define those that should be present in every font (basically T1). This inheritance list should be overridden by font-specific ones.

```

5198 \DeclareCharacterInheritance
5199 { encoding = {TU,EU1,EU2} }
5200 { A = {\`A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
5201   a = {\`a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
5202   C = {\'C,\c C,\v C},
5203   c = {\'c,\c c,\v c},
5204   D = {\v D,\DH},
5205   d = {\v d,\dj},
5206   E = {\`E,\'E,\^E,\"E,\k E,\v E},
5207   e = {\`e,\'e,\^e,\"e,\k e,\v e},
5208   f = {/f_f}, % sometimes /f_f, sometimes /ff
5209   G = {\u G},
5210   g = {\u g},
5211   I = {\`I,\'I,\^I,\"I,\.I},
5212   i = {\`i,\'i,\^i,\"i,\i},
5213   j = {\j},
5214   L = {\L,\'L,\v L},
5215   l = {\l,\'l,\v l},
5216   N = {\'N,\-N,\v N},
5217   n = {\'n,\-n,\v n},
5218   O = {\0,\`0,\'0,\^0,\-0,\"0,\H 0},
5219   o = {\o,\`o,\'o,\^o,\-o,\"o,\H o},
5220   R = {\'R,\v R},
5221   r = {\'r,\v r},
5222   S = {\'S,\c S,\v S}, % \SS
5223   s = {\'s,\c s,\v s},
5224   T = {\c T,\v T},
5225   t = {\c t,\v t},
5226   U = {\`U,\'U,\^U,\"U,\H U,\r U},
5227   u = {\`u,\'u,\^u,\"u,\H u,\r u},
5228   Y = {\'Y,\"Y},
5229   y = {\'y,\"y},
5230   Z = {\'Z,\.Z,\v Z},
5231   z = {\'z,\.z,\v z}
5232 }
5233
5234 </m-t>

```

### 15.5.8 Euro symbols

Make Euro symbols settings simpler.

```

5235 <*zpeu>
5236 \DeclareCharacterInheritance
5237 { encoding = U,
5238   family   = {zpeu,zpeus,eurosans} }
5239 { E = 128 }
5240
5241 </zpeu>
5242 <mv>

```

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.

```

5243 \DeclareCharacterInheritance
5244 { encoding = {OT1,U},

```

```

5245     family    = mvs }
5246     { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
5247
5248 </mvs>

```

## 15.6 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.

```

5249 <*m-t>
5250 %%% -----
5251 %%% TRACKING/LETTERSPACING
5252
5253 \SetTracking
5254 [ name          = default,
5255   no ligatures = {f} ]
5256 { encoding      = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
5257 { }
5258

```

## 15.7 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).

```

5259 %%% -----
5260 %%% EXPANSION
5261
5262 \SetExpansion
5263 [ name          = default ]
5264 { encoding = {OT1,OT4,QX,T1,LY1} }
5265 {
5266   A = 500,      a = 700,
5267   \AE = 500,    \ae = 700,
5268   B = 700,      b = 700,
5269   C = 700,      c = 700,
5270   D = 500,      d = 700,
5271   E = 700,      e = 700,
5272   F = 700,
5273   G = 500,      g = 700,
5274   H = 700,      h = 700,
5275   K = 700,      k = 700,
5276   M = 700,      m = 700,
5277   N = 700,      n = 700,
5278   O = 500,      o = 700,
5279   \OE = 500,    \oe = 700,
5280   P = 700,      p = 700,
5281   Q = 500,      q = 700,
5282   R = 700,
5283   S = 700,      s = 700,
5284   U = 700,      u = 700,
5285   W = 700,      w = 700,
5286   Z = 700,      z = 700,
5287   2 = 700,
5288   3 = 700,
5289   6 = 700,
5290   8 = 700,
5291   9 = 700
5292 }
5293

```



Settings for Cyrillic T2A encoding.<sup>19</sup>

```

5294 \SetExpansion
5295   [ name      = T2A ]
5296   { encoding = T2A }
5297   {
5298     A = 500,      a = 700,
5299     B = 700,      b = 700,
5300     C = 700,      c = 700,
5301     D = 500,      d = 700,
5302     E = 700,      e = 700,
5303     F = 700,
5304     G = 500,      g = 700,
5305     H = 700,      h = 700,
5306     K = 700,      k = 700,
5307     M = 700,      m = 700,
5308     N = 700,      n = 700,
5309     O = 500,      o = 700,
5310     P = 700,      p = 700,
5311     Q = 500,      q = 700,
5312     R = 700,
5313     S = 700,      s = 700,
5314     U = 700,      u = 700,
5315     W = 700,      w = 700,
5316     Z = 700,      z = 700,
5317     2 = 700,
5318     3 = 700,
5319     6 = 700,
5320     8 = 700,
5321     9 = 700,
5322     \CYRA = 500,   \cyra = 700,
5323     \CYRB = 700,   \cyrb = 700,
5324     \CYRV = 700,   \cyrv = 700,
5325     \CYRG = 700,   \cyrg = 700,
5326     \CYRD = 700,   \cyrd = 700,
5327     \CYRE = 700,   \cyre = 700,
5328     \CYRZH = 700,  \cyrzh = 700,
5329     \CYRZ = 700,   \cyrz = 700,
5330     \CYRI = 700,   \cyri = 700,
5331     \CYRISHRT = 700, \cyrishrt = 700,
5332     \CYRK = 700,   \cyrk = 700,
5333     \CYRL = 700,   \cyrl = 700,
5334     \CYRM = 700,   \cyrm = 700,
5335     \CYRN = 700,   \cyrn = 700,
5336     \CYRO = 500,   \cyro = 700,
5337     \CYRP = 700,   \cyrp = 700,
5338     \CYRR = 700,   \cyrr = 700,
5339     \CYRS = 700,   \cyrs = 700,
5340     \CYRT = 700,   \cyrt = 700,
5341     \CYRU = 700,   \cyru = 700,
5342     \CYRF = 700,   \cyrf = 700,
5343     \CYRH = 700,   \cyrh = 700,
5344     \CYRC = 700,   \cyrc = 700,
5345     \CYRCH = 700,  \cyrch = 700,
5346     \CYRSH = 700,  \cyrsh = 700,
5347     \CYRSHCH = 700, \cyrshch = 700,
5348     \CYRHRDSN = 700, \cyhrdsn = 700,
5349     \CYRERY = 700,  \cyrery = 700,
5350     \CYRSFTSN = 700, \cysftsn = 700,
5351     \CYREREV = 700, \cyrerev = 700,
5352     \CYRYU = 700,   \cyryu = 700,
5353     \CYRYA = 700,   \cyrya = 700
5354   }
5355

```

---

19 Contributed by *Karl Karlsson*.

T5 encoding does not contain \AE, \ae, \OE and \oe.

```

5356 \SetExpansion
5357   [ name      = T5 ]
5358   { encoding = T5 }
5359   {
5360     A = 500,      a = 700,
5361     B = 700,      b = 700,
5362     C = 700,      c = 700,
5363     D = 500,      d = 700,
5364     E = 700,      e = 700,
5365     F = 700,
5366     G = 500,      g = 700,
5367     H = 700,      h = 700,
5368     K = 700,      k = 700,
5369     M = 700,      m = 700,
5370     N = 700,      n = 700,
5371     O = 500,      o = 700,
5372     P = 700,      p = 700,
5373     Q = 500,      q = 700,
5374     R = 700,
5375     S = 700,      s = 700,
5376     U = 700,      u = 700,
5377     W = 700,      w = 700,
5378     Z = 700,      z = 700,
5379     2 = 700,
5380     3 = 700,
5381     6 = 700,
5382     8 = 700,
5383     9 = 700
5384   }
5385
5386 /m-t

```

## 15.8 Character protrusion

```

5387 %%% -----
5388 %%% PROTRUSION
5389

```

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}
}

```

### 15.8.1 Normal

The default settings always use the most moderate value.

```

5390 <*cfg-t>
5391 \SetProtrusion
5392 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```

5393 <bch> [ name = bch-default ]

```

- Bitstream Letter Gothic (blg)

```

5394 <blg> [ name = blg-default ]

```

- Computer Modern Roman (cmr)

```

5395 <cmr> [ name = cmr-default ]

```

- EB Garamond

```

5396 <ebg> [ name = EBGaramond-default ]

```

- Minion<sup>20</sup> (pmnx, pmnj)

```

5397 <pmn> [ name = pmnj-default ]

```

- Palatino (ppl, pplx, pplj)

```

5398 <ppl> [ name = ppl-default ]

```

- Times (ptm, ptmx, ptmj)

```

5399 <ptm> [ name = ptm-default ]

```

- URW Garamond (ugm)

```

5400 <ugm> [ name = ugm-default ]
5401 <m-t|cmr|pmn> { }
5402 <bch|blg|ebg|ugm> { encoding = OT1,
5403 <ppl|ptm> { encoding = {OT1,OT4},
5404 <bch> family = bch }
5405 <blg> family = blg }
5406 <ebg> family = {EBGaramond-LF,EBGaramond-Of,EBGaramond-TLF,EBGaramond-TOf} }
5407 <ppl> family = {ppl,pplx,pplj} }
5408 <ptm> family = {ptm,ptmx,ptmj} }
5409 <ugm> family = ugm }
5410 {
5411 <m-t|bch|blg|cmr|ebg|pmn|ppl|ptm> A = {50,50},
5412 <ugm> A = {50,100},
5413 <ebg|ptm> \AE = {50, },
5414 <ugm> \AE = {150,50},
5415 <ugm> B = { ,50},
5416 <bch|ebg|pmn|ugm> C = {50, },

```

---

20 Contributed by Harald Harders and Karl Karlsson.

```

5417 <bch|ebg|pmn>    D = { ,50},
5418 <ugm>            D = { ,70},
5419 <ugm>            E = { ,50},
5420 <m-t|bch|cmr|ebg|pmn|ptm>    F = { ,50},
5421 <ugm>            F = { ,70},
5422 <bch|ebg|pmn>    G = {50, },
5423 <ugm>            G = {50,50},
5424 <blg>            I = {150,150},
5425 <m-t|cmr|ebg|pmn|ppl|ptm|ugm>    J = {50, },
5426 <bch|blg>        J = {100, },
5427 <!blg>          K = { ,50},
5428 <blg>            K = {50, },
5429 <m-t|bch|cmr|ebg|pmn|ppl>    L = { ,50},
5430 <blg>            L = { ,150},
5431 <ptm>            L = { ,80},
5432 <ugm>            L = { ,120},
5433 <bch|ebg|pmn|ugm>    O = {50,50},
5434 <ebg>            \OE = {50, },
5435 <ugm>            \OE = {50,50},
5436 <blg>            P = { ,100},
5437 <ugm>            P = { ,50},
5438 <bch|ebg|pmn>    Q = {50,70},
5439 <ugm>            Q = {50,50},
5440 <bch>            R = { ,50},
5441 <ugm>            R = { ,70},
5442 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    T = {50,50},
5443 <blg>            T = {100,100},
5444 <ugm>            T = {70,70},
5445 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    V = {50,50},
5446 <blg|ugm>        V = {70,70},
5447 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    W = {50,50},
5448 <ugm>            W = {70,70},
5449 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    X = {50,50},
5450 <ugm>            X = {50,70},
5451 <m-t|bch|cmr|ebg|pmn|ppl>    Y = {50,50},
5452 <blg|ptm|ugm>    Y = {80,80},
5453 <ugm>            Z = {50,50},
5454 <blg>            f = {150,100},
5455 <blg>            i = {150,150},
5456 <blg>            j = {100,100},
5457 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    k = { ,50},
5458 <ugm>            k = { ,70},
5459 <blg>            l = {150,150},
5460 <pmn>            l = { , -50},
5461 <ebg|ppl>        p = {50,50},
5462 <ugm>            p = { ,50},
5463 <ebg|ppl>        q = {50, },
5464 <!blg>          r = { ,50},
5465 <blg>            r = {100, 80},
5466 <cmr|ebg|pmn>    t = { ,70},
5467 <bch>            t = { ,50},
5468 <blg>            t = {150, 80},
5469 <ugm>            t = { ,100},
5470 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    v = {50,50},
5471 <blg>            v = {100,100},
5472 <ugm>            v = {50,70},
5473 <m-t|bch|cmr|ebg|pmn|ppl|ptm>    w = {50,50},
5474 <ugm>            w = {50,70},
5475 <!blg>          x = {50,50},
5476 <blg>            x = {100,100},
5477 <m-t|bch|ebg|pmn>    y = { ,50},
5478 <blg>            y = { 50,100},
5479 <cmr|ppl|ptm>    y = {50,70},
5480 <ugm>            y = { ,70},
5481 <cmr>            O = { ,50},

```

```

5482 <m-t> 1 = {50,50},
5483 <bch|blg|ebg|ptm|ugm> 1 = {150,150},
5484 <cmr> 1 = {100,200},
5485 <pmn> 1 = { ,50},
5486 <ppl> 1 = {100,100},
5487 <bch|cmr|ebg|ugm> 2 = {50,50},
5488 <blg> 2 = { ,100},
5489 <bch|pmn> 3 = {50, },
5490 <cmr|ebg|ugm> 3 = {50,50},
5491 <blg> 3 = {100, },
5492 <m-t|ebg> 4 = {50,50},
5493 <bch> 4 = {100,50},
5494 <blg> 4 = {100, },
5495 <cmr|ugm> 4 = {70,70},
5496 <pmn> 4 = {50, },
5497 <ptm> 4 = {70, },
5498 <cmr> 5 = { ,50},
5499 <ebg> 5 = {50,50},
5500 <bch> 6 = {50, },
5501 <cmr> 6 = { ,50},
5502 <ebg> 6 = {50,50},
5503 <m-t> 7 = {50,50},
5504 <bch|ebg|pmn|ugm> 7 = {50,80},
5505 <blg> 7 = {100,100},
5506 <cmr|ptm> 7 = {50,100},
5507 <ppl> 7 = { ,50},
5508 <cmr> 8 = { ,50},
5509 <bch|ebg> 9 = {50,50},
5510 <cmr> 9 = { ,50},
5511 <m-t|cmr|ebg|pmn|ppl|ptm|ugm> . = { ,700},
5512 <bch> . = { ,600},
5513 <blg> . = {400,500},
5514 <!blg> {,}= { ,500},
5515 <blg> {,}= {300,400},
5516 <m-t|cmr|ebg|pmn|ppl|ptm|ugm> : = { ,500},
5517 <bch> : = { ,400},
5518 <blg> : = {300,400},
5519 <m-t|bch|ebg|pmn|ptm> ; = { ,300},
5520 <blg> ; = {200,300},
5521 <cmr|ppl> ; = { ,500},
5522 <ugm> ; = { ,400},
5523 <!blg> ! = { ,100},
5524 <blg> ! = {200,200},
5525 <m-t|ebg|pmn|ptm> ? = { ,100},
5526 <bch|cmr|ppl|ugm> ? = { ,200},
5527 <blg> ? = {150,150},
5528 <pmn> " = {300,300},
5529 <m-t|bch|cmr|ebg|pmn|ppl> @ = {50,50},
5530 <ptm> @ = {100,100},
5531 <m-t|bch|blg|cmr|ebg|pmn|ppl|ptm> ~ = {200,250},
5532 <ugm> ~ = {300,350},
5533 <ebg|ppl|ptm> & = {50,100},
5534 <ugm> & = { ,100},
5535 <m-t|cmr|ebg|pmn> \% = {50,50},
5536 <bch> \% = { ,50},
5537 <ppl|ptm> \% = {100,100},
5538 <ugm> \% = {50,100},
5539 <blg> \# = {100,100},
5540 <m-t|ppl|ptm|ugm> * = {200,200},
5541 <bch|pmn> * = {200,300},
5542 <blg> * = {150,200},
5543 <cmr|ebg> * = {300,300},
5544 <m-t|cmr|ppl|ptm> + = {250,250},
5545 <bch> + = {150,250},
5546 <ebg> + = {300,300},

```

```

5547 <blg|pmn>      + = {150,200},
5548 <ugm>          + = {250,300},
5549 <blg|ugm>      {=} = {200,200},
5550 <m-t|ebg|pmn|ptm> ( = {100,   },   ) = {   ,200},
5551 <bch|ugm>        ( = {200,   },   ) = {   ,200},
5552 <cmr|blg>        ( = {300,   },   ) = {   ,300},
5553 <ppl>            ( = {100,   },   ) = {   ,300},
5554 <bch|pmn>        [ = {100,   },   ] = {   ,100},
5555 <blg>           [ = {300,100},   ] = {   ,300},

5556 <m-t|ebg|pmn|ptm> / = {100,200},
5557 <bch>           / = {   ,200},
5558 <blg>           / = {300,300},
5559 <cmr|ppl>       / = {200,300},
5560 <ugm>           / = {100,300},
5561 <m-t|ptm>       - = {500,500},
5562 <bch|cmr|ppl>   - = {400,500},
5563 <blg>           - = {300,400},
5564 <ebg>           - = {300,500},
5565 <pmn>           - = {200,400},
5566 <ugm>           - = {500,600},
5567 <blg>           < = {200,100},   > = {100,200},
5568 <blg>           _ = {150,250},
5569 <blg>           | = {250,250},
5570 <m-t|pmn>       \textendash      = {200,200},   \textendash      = {150,150},
5571 <bch>           \textendash      = {200,300},   \textendash      = {150,250},
5572 <cmr>           \textendash      = {400,300},   \textendash      = {300,200},
5573 <ebg|ppl|ptm>   \textendash      = {300,300},   \textendash      = {200,200},
5574 <ugm>           \textendash      = {250,300},   \textendash      = {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).

```

5575 <m-t|bch|pmn>   \textquoteleft   = {300,400},   \textquoteright = {300,400},
5576 <blg>           \textquoteleft   = {400,600},   \textquoteright = {400,600},
5577 <cmr>           \textquoteleft   = {500,700},   \textquoteright = {500,600},
5578 <ebg|ppl>       \textquoteleft   = {500,700},   \textquoteright = {500,700},
5579 <ptm>           \textquoteleft   = {500,500},   \textquoteright = {300,500},
5580 <ugm>           \textquoteleft   = {300,600},   \textquoteright = {300,600},
5581 <m-t|bch|pmn>   \textquotedblleft = {300,300},   \textquotedblright = {300,300}
5582 <blg>           \textquotedblright = {300,400}
5583 <cmr>           \textquotedblleft = {500,300},   \textquotedblright = {200,600}
5584 <ebg|ppl|ptm>   \textquotedblleft = {300,400},   \textquotedblright = {300,400}
5585 <ugm>           \textquotedblleft = {400,400},   \textquotedblright = {400,400}
5586 }
5587

```

Greek uppercase letters are in OT1 encoding only.

```

5588 <*m-t|cmr|pmn>
5589 \SetProtrusion
5590 <m-t> [ name      = OT1-default,
5591 <cmr> [ name      = cmr-OT1,
5592 <pmn> [ name      = pmnj-OT1,
5593 <m-t> [ load       = default ]
5594 <cmr> [ load       = cmr-default ]
5595 <pmn> [ load       = pmnj-default ]
5596 <m-t> { encoding = OT1 }
5597 <cmr> { encoding = {OT1,OT4},
5598 <pmn> { encoding = OT1,
5599 <cmr>   family   = cmr }
5600 <pmn>   family   = pmnj }
5601 {
5602 <m-t|cmr> \AE = {50,   },
5603 <pmn>     \OE = {50,   }
5604 <*cmr>
5605 "00 = {   ,150}, % \Gamma

```

```

5606 "01 = {100,100}, % \Delta
5607 "02 = { 50, 50}, % \Theta
5608 "03 = {100,100}, % \Lambda
5609 "06 = { 50, 50}, % \Sigma
5610 "07 = {100,100}, % \Upsilon
5611 "08 = { 50, 50}, % \Phi
5612 "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

5613 </cmr>
5614 }
5615
5616 </m-t|cmr|pmn>

```

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For X<sub>Y</sub>TeX (EU1) and LuaTeX (EU2) we simply use the T1 list as default (for now).

```

5617 \SetProtrusion
5618 <m-t> [ name = T1-default,
5619 <bch> [ name = bch-T1,
5620 <blg> [ name = blg-T1,
5621 <cmr> [ name = cmr-T1,
5622 <ebg> [ name = EBGaramond-T1,
5623 <pmn> [ name = pmnj-T1,
5624 <ppl> [ name = ppl-T1,
5625 <ptm> [ name = ptm-T1,
5626 <ugm> [ name = ugm-T1,
5627 <m-t> load = default ]
5628 <bch> load = bch-default ]
5629 <blg> load = blg-default ]
5630 <cmr> load = cmr-default ]
5631 <ebg> load = EBGaramond-default ]
5632 <pmn> load = pmnj-default ]
5633 <ppl> load = ppl-default ]
5634 <ptm> load = ptm-default ]
5635 <ugm> load = ugm-default ]
5636 <m-t> { encoding = {T1,LY1,EU1,EU2,TU} }
5637 <bch|cmr|ebg|pmn|ppl> { encoding = {T1,LY1},
5638 <blg|ptm|ugm> { encoding = {T1},
5639 <bch> family = bch }
5640 <blg> family = blg }
5641 <cmr> family = cmr }
5642 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOf} }
5643 <pmn> family = pmnj }
5644 <ppl> family = {ppl,pplx,pplj} }
5645 <ptm> family = {ptm,ptmx,ptmj} }
5646 <ugm> family = ugm }
5647 {
5648 <m-t|cmr> \AE = {50, },
5649 <bch|pmn> \OE = {50, },
5650 <pmn> \TH = { ,50},
5651 <blg> \v L = { ,250},
5652 <blg> \v d = { ,250},
5653 <blg> \v l = { ,250},
5654 <blg> \v t = { ,250},
5655 <blg> 127 = {300,400},
5656 <blg> 156 = {100, }, % IJ
5657 <blg> 188 = { 80, 80}, % ij
5658 <m-t|bch|ebg|pmn|ppl|ptm> _ = {100,100},
5659 <cmr> _ = {200,200},
5660 <ugm> _ = {100,200},
5661 <m-t|ebg|pmn|ptm> \textbackslash = {100,200},
5662 <bch> \textbackslash = {150,200},
5663 <blg> \textbackslash = {250,300},
5664 <cmr|ppl> \textbackslash = {200,300},

```

```

5665 <ugm> \textbackslash = {100,300},
5666 <ugm> \textbar = {200,200},
5667 <blg> \textendash = {300,300}, \textemdash = {150,150},
5668 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
5669 <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.

```

5670 <m-t|cmr|ebg|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5671 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
5672 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5673 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
5674 <blg> \guilsinglleft = {300,500}, \guilsinglright = {300,500},
5675 <cmr|ebg|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5676 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5677 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5678 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5679 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
5680 <blg|ebg|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5681 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
5682 <m-t|bch|cmr|ebg|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
5683 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
5684 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
5685 <m-t|cmr|ebg|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
5686 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
5687 <m-t|bch|cmr|ebg|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
5688 <pmn> \textless = {100, }, \textgreater = { ,100},
5689 <pmn> \textvisiblespace = {100,100} % not in LY1
5690 }
5691

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

5692 <*cmr>
5693 \SetProtrusion
5694 [ name = lmr-T1,
5695   load = cmr-T1 ]
5696 { encoding = {T1,LY1},
5697   family = lmr }
5698 {
5699   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
5700 }
5701
5702 </cmr>

```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).<sup>21</sup>

```

5703 <*m-t|cmr|pmn>
5704 \SetProtrusion
5705 <m-t> [ name = T2A-default,
5706 <cmr> [ name = cmr-T2A,
5707 <pmn> [ name = pmnj-T2A,
5708 <m-t> load = default ]
5709 <cmr> load = cmr-default ]
5710 <pmn> load = pmnj-default ]
5711 { encoding = T2A,
5712 <m-t> }
5713 <cmr> family = cmr }
5714 <pmn> family = pmnj }
5715 {
5716   \CYRA = {50,50},
5717   \CYRG = { ,50},
5718   \CYRK = { ,50},

```

21 Contributed by Karl Karlsson.



```

5719 \CYRT = {50,50},
5720 \CYRH = {50,50},
5721 \CYRU = {50,50},
5722 <pmn> \CYRS = {50, },
5723 <pmn> \CYRO = {50,50},
5724 \cyrk = { ,50},
5725 \cyrp = { ,50},
5726 \cyrh = {50,50},
5727 <m-t|pmn> \cyru = {50,50},
5728 <cmr> \cyru = {50,70},
5729 <m-t> _ = {100,100},
5730 <cmr> _ = {200,200},
5731 <m-t> \textbackslash = {100,200}, \quotedblbase = {400,400},
5732 <cmr> \textbackslash = {200,300}, \quotedblbase = {400,400},
5733 <pmn> \textbackslash = {100,200}, \quotedblbase = {300,300},
5734 <cmr> \textquotedbl = {300,300}, \textquotedblleft = {200,600},
5735 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5736 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5737 <pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
5738 <m-t|cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
5739 <pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
5740 <m-t|cmr> \textless = {200,100}, \textgreater = {100,200}
5741 <pmn> \textless = {100, }, \textgreater = { ,100}
5742 }
5743
5744 </m-t|cmr|pmn>

```

Settings for the QX encoding (generic and Times).<sup>22</sup> It also includes some glyphs otherwise in TS1.

```

5745 <*m-t|ptm>
5746 \SetProtrusion
5747 <m-t> [ name = QX-default,
5748 <ptm> [ name = ptm-QX,
5749 <m-t> load = default ]
5750 <ptm> load = ptm-default ]
5751 <m-t> { encoding = QX }
5752 <ptm> { encoding = QX,
5753 <ptm> family = {ptm,ptmx,ptmj} }
5754 {
5755 \AE = {50, },
5756 <ptm> * = {200,200},
5757 {=} = {100,100},
5758 \textunderscore = {100,100},
5759 \textbackslash = {100,200},
5760 \quotedblbase = {400,400},
5761 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5762 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5763 \textexclamdown = {100, }, \textquestiondown = {100, },
5764 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
5765 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
5766 \textless = {200,100}, \textgreater = {100,200},
5767 \textminus = {200,200}, \textdegree = {300,300},
5768 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5769 <ptm> \copyright = {100,150}, \textregistered = {100,150},
5770 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
5771 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
5772 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
5773 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
5774 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
5775 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5776 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5777 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
5778 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},

```

```

5779 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
5780 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5781 <ptm> \textperthousand = { ,50}
5782 }
5783
5784 </m-t|ptm>

```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented characters are already taken care of by the inheritance list.

```

5785 <*cmr|bch>
5786 \SetProtrusion
5787 <cmr> [ name = cmr-T5,
5788 <cmr> load = cmr-default ]
5789 <bch> [ name = bch-T5,
5790 <bch> load = bch-default ]
5791 { encoding = T5,
5792 <cmr> family = cmr }
5793 <bch> family = bch }
5794 {
5795 <bch> _ = {100,100},
5796 <bch> \textbackslash = {150,200},
5797 <cmr> \textbackslash = {200,300},
5798 <cmr> \textquotedblleft = {200,600},
5799 <cmr> \textquotedbl = {300,300},
5800 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5801 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5802 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
5803 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5804 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
5805 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5806 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
5807 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
5808 \textless = {200,100}, \textgreater = {100,200}
5809 }
5810
5811 </cmr|bch>

```

Minion with lining numbers.

```

5812 <*pmn>
5813 \SetProtrusion
5814 [ name = pmnx-OT1,
5815 load = pmnj-default ]
5816 { encoding = OT1,
5817 family = pmnx }
5818 {
5819 1 = {230,180}
5820 }
5821
5822 \SetProtrusion
5823 [ name = pmnx-T1,
5824 load = pmnj-T1 ]
5825 { encoding = {T1,LY1},
5826 family = pmnx }
5827 {
5828 1 = {230,180}
5829 }
5830
5831 \SetProtrusion
5832 [ name = pmnx-T2A,
5833 load = pmnj-T2A ]
5834 { encoding = {T2A},
5835 family = pmnx }
5836 {
5837 1 = {230,180}
5838 }

```

```
5839
5840 </pmn>
```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```
5841 <*ptm>
5842 \SetProtrusion
5843 [ name      = ptm-LY1,
5844   load      = ptm-T1 ]
5845 { encoding = LY1,
5846   family   = {ptm,ptmx,ptmj} }
5847 {
5848   _                      = {100,100},
5849   \texttrademark         = {100,100},
5850   \textregistered       = {100,100},
5851   \textcopyright        = {100,100},
5852   \textdegree           = {300,300},
5853   \textminus            = {200,200},
5854   \textellipsis         = {150,200},
5855   \% \texteuro          = { , }, % ?
5856   \textcent             = {100,100},
5857   \textquotesingle      = {500,500},
5858   \textflorin           = { 50, 70},
5859   \textdagger           = {150,150},
5860   \textdaggerdbl        = {100,100},
5861   \textperthousand      = { , 50},
5862   \textbullet           = {150,150},
5863   \textonesuperior      = {100,100},
5864   \texttwosuperior      = { 50, 50},
5865   \textthreesuperior    = { 50, 50},
5866   \textperiodcentered   = {300,300},
5867   \textplusminus        = { 50, 80},
5868   \textmultiply         = {100,100},
5869   \textdivide           = { 50,150}
```

Remaining slots in the source file.

```
5870 }
5871
5872 </ptm>
```

## 15.8.2 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.<sup>23</sup>

```
5873 \SetProtrusion
5874 <m-t> [ name      = OT1-it  ]
5875 <bch> [ name      = bch-it  ]
5876 <blg> [ name      = blg-it,
5877 <blg>   load      = blg-default ]
5878 <cmr> [ name      = cmr-it  ]
5879 <ebg> [ name      = EBGaramond-it ]
5880 <pmn> [ name      = pmnj-it ]
5881 <ppl> [ name      = ppl-it  ]
5882 <ptm> [ name      = ptm-it  ]
5883 <ugm> [ name      = ugm-it  ]
5884 <m-t|bch|blg|ebg|ugm> { encoding = OT1,
```

---

23 Settings contributed by Hendrik Vogt.

```

5885 < ppl | ptm > { encoding = {OT1,OT4},
5886 < bch > family = bch,
5887 < blg > family = blg,
5888 < ebg > family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF},
5889 < ppl > family = {ppl,pplx,pplj},
5890 < ptm > family = {ptm,ptmx,ptmj},
5891 < ugm > family = ugm,
5892 < m-t | bch | ebg | ppl | ptm > shape = {it,sl} }
5893 < blg | ugm > shape = it }
5894 < cmr | pmn > { }
5895 {
5896 < cmr > A = {100,100},
5897 < ptm > A = {100,50},
5898 < ebg | pmn > A = {50, },
5899 < ugm > A = { ,150},
5900 < ppl > A = {50,50},
5901 < ptm > \AE = {100, },
5902 < ebg | ppl > \AE = {50, },
5903 < cmr > B = {83,-40},
5904 < ebg | ppl | ptm > B = {50, },
5905 < pmn > B = {20,-50},
5906 < bch | ppl | ptm | ugm > C = {50, },
5907 < cmr > C = {165,-75},
5908 < ebg > C = {100, },
5909 < pmn > C = {50,-50},
5910 < cmr > D = {75, -28},
5911 < ebg | ppl | ptm > D = {50,50},
5912 < pmn > D = {20, },
5913 < cmr > E = {80,-55},
5914 < ebg | ppl | ptm > E = {50, },
5915 < pmn > E = {20,-50},
5916 < cmr > F = {85,-80},
5917 < ebg | ptm > F = {100, },
5918 < pmn > F = {10, },
5919 < ppl > F = {50, },
5920 < bch | ppl | ptm | ugm > G = {50, },
5921 < cmr > G = {153,-15},
5922 < ebg > G = {100, },
5923 < pmn > G = {50,-50},
5924 < cmr > H = {73,-60},
5925 < ebg | ppl | ptm > H = {50, },
5926 < cmr > I = {140,-120},
5927 < ebg | ptm > I = {50, },
5928 < pmn > I = {20,-50},
5929 < cmr > J = {135,-80},
5930 < ebg > J = {50, },
5931 < pmn > J = {20, },
5932 < ptm > J = {100, },
5933 < cmr > K = {70,-30},
5934 < ebg | ppl | ptm > K = {50, },
5935 < pmn > K = {20, },
5936 < cmr > L = {87, 40},
5937 < ebg | ppl | ptm > L = {50, },
5938 < pmn > L = {20,50},
5939 < ugm > L = { ,100},
5940 < cmr > M = {67,-45},
5941 < pmn > M = { , -30},
5942 < ptm > M = {50, },
5943 < cmr > N = {75,-55},
5944 < pmn > N = { , -30},
5945 < ptm > N = {50, },
5946 < bch | pmn | ppl | ptm > O = {50, },
5947 < cmr > O = {150,-30},
5948 < ebg > O = {100, },
5949 < ugm > O = {70,50},

```

```

5950 <ppl|ptm> \OE = {50, },
5951 <ebg> \OE = {100, },
5952 <cmr> P = {82,-50},
5953 <ebg|ppl|ptm> P = {50, },
5954 <pmn> P = {20,-50},
5955 <bch|pmn|ppl|ptm> Q = {50, },
5956 <cmr> Q = {150,-30},
5957 <ebg> Q = {100, },
5958 <ugm> Q = {70,50},
5959 <cmr> R = {75, 15},
5960 <ebg|ppl|ptm> R = {50, },
5961 <pmn> R = {20, },
5962 <bch|ebg|ppl|ptm> S = {50, },
5963 <cmr> S = {90,-65},
5964 <pmn> S = {20,-30},
5965 <bch|ebg|ppl|ptm> $ = {50, },
5966 <cmr> $ = {100,-20},
5967 <pmn> $ = {20,-30},
5968 <bch|pmn|ugm> T = {70, },
5969 <cmr> T = {220,-85},
5970 <ebg|ppl|ptm> T = {100, },
5971 <cmr> U = {230,-55},
5972 <ebg|ppl|ptm> U = {50, },
5973 <pmn> U = {50,-50},
5974 <cmr> V = {260,-60},
5975 <ebg|pmn|ugm> V = {100, },
5976 <ppl|ptm> V = {100,50},
5977 <cmr> W = {185,-55},
5978 <ebg|pmn|ugm> W = {100, },
5979 <ppl> W = {50, },
5980 <ptm> W = {100,50},
5981 <cmr> X = {70,-30},
5982 <ppl|ptm> X = {50, },
5983 <cmr> Y = {250,-60},
5984 <pmn> Y = {50, },
5985 <ppl> Y = {100,50},
5986 <ptm> Y = {100, },
5987 <cmr> Z = {90,-60},
5988 <pmn> Z = { , -50},
5989 <cmr> a = {150,-10},
5990 <cmr> b = {170, },
5991 <cmr> c = {173,-10},
5992 <cmr> d = {150,-55},
5993 <pmn> d = { , -50},
5994 <cmr> e = {180, },
5995 <cmr> f = { , -250},
5996 <ebg|pmn> f = { , -100},
5997 <cmr> g = {150,-10},
5998 <cmr> h = {100, },
5999 <cmr> i = {210, },
6000 <pmn> i = { , -30},
6001 <cmr> j = { , -40},
6002 <pmn> j = { , -30},
6003 <cmr> k = {110,-50},
6004 <cmr> l = {240,-110},
6005 <pmn> l = { , -100},
6006 <cmr> m = {80, },
6007 <cmr> n = {115, },
6008 <bch> o = {50,50},
6009 <cmr> o = {155, },
6010 <bch> p = { , 50},
6011 <pmn> p = {-50, },
6012 <bch> q = {50, },
6013 <cmr> q = {170,-40},
6014 <cmr> r = {155,-40},

```

```

6015 <pmn>      r = { ,50},
6016 <cmr>      s = {130, },
6017 <bch>      t = { ,50},
6018 <cmr>      t = {230,-10},
6019 <cmr>      u = {120, },
6020 <cmr>      v = {140,-25},
6021 <pmn|ugm>  v = {50, },
6022 <bch>      w = { ,50},
6023 <cmr>      w = {98,-20},
6024 <pmn|ugm>  w = {50, },
6025 <cmr>      x = {65,-40},
6026 <bch>      y = { ,50},
6027 <cmr>      y = {130,-20},
6028 <cmr>      z = {110,-80},
6029 <cmr>      0 = {170,-85},
6030 <bch|ptm>  1 = {150,100},
6031 <cmr>      1 = {230,110},
6032 <ebg>      1 = {150, },
6033 <pmn>      1 = {50, },
6034 <ppl>      1 = {100, },
6035 <ugm>      1 = {150,150},
6036 <cmr>      2 = {130,-70},
6037 <ebg|ppl|ptm> 2 = {50, },
6038 <pmn>      2 = {-50, },
6039 <bch>      3 = {50, },
6040 <cmr>      3 = {140,-70},
6041 <pmn>      3 = {-100, },
6042 <ptm>      3 = {100,50},
6043 <bch>      4 = {100, },
6044 <cmr>      4 = {130,80},
6045 <ebg>      4 = {150, },
6046 <ppl|ptm> 4 = {50, },
6047 <cmr>      5 = {160, },
6048 <ptm>      5 = {50, },
6049 <bch>      6 = {50, },
6050 <cmr>      6 = {175,-30},
6051 <bch|ebg|ptm> 7 = {100, },
6052 <cmr>      7 = {250,-150},
6053 <pmn>      7 = {20, },
6054 <ppl>      7 = {50, },
6055 <cmr>      8 = {130,-40},
6056 <cmr>      9 = {155,-80},
6057 <m-t|cmr|ebg|pmn|ppl> . = { ,500},
6058 <blg>      . = {400,600},
6059 <bch|ptm|ugm> . = { ,700},
6060 <blg>      {,}= {300,500},
6061 <m-t|ebg|pmn|ppl> {,}= { ,500},
6062 <cmr>      {,}= { ,450},
6063 <bch|ugm>   {,}= { ,600},
6064 <ptm>      {,}= { ,700},
6065 <m-t|cmr|ebg|ppl> : = { ,300},
6066 <bch|ugm>   : = { ,400},
6067 <pmn>      : = { ,200},
6068 <ptm>      : = { ,500},
6069 <m-t|cmr|ebg|ppl> ; = { ,300},
6070 <bch|ugm>   ; = { ,400},
6071 <pmn>      ; = { ,200},
6072 <ptm>      ; = { ,500},
6073 <ptm>      ! = { ,100},
6074 <bch>      ? = { ,200},
6075 <ptm>      ? = { ,100},
6076 <ppl>      ? = { ,300},
6077 <pmn>      " = {400,200},
6078 <m-t|ebg|pmn|ppl|ptm> & = {50,50},
6079 <bch>      & = { ,80},

```

```

6080 <cmr>      & = {130,30},
6081 <ugm>      & = {50,100},
6082 <m-t|ebg|pmn>    \% = {100, },
6083 <cmr>      \% = {180,50},
6084 <bch>      \% = {50,50},
6085 <ppl|ptm>    \% = {100,100},
6086 <ugm>      \% = {100,50},
6087 <m-t|pmn|ppl>    * = {200,200},
6088 <bch>      * = {300,200},
6089 <cmr>      * = {380,20},
6090 <ebg>      * = {500,100},
6091 <ptm|ugm>    * = {400,200},
6092 <m-t|pmn|ppl>    + = {150,200},
6093 <cmr>      + = {180,200},
6094 <bch|ugm>    + = {250,250},
6095 <ebg|ptm>    + = {250,200},
6096 <m-t|ebg|pmn|ppl> @ = {50,50},
6097 <bch>      @ = {80,50},
6098 <cmr>      @ = {180,10},
6099 <ptm>      @ = {150,150},
6100 <m-t|bch|ugm>   ~ = {150,150},
6101 <cmr|ebg|pmn|ppl|ptm> ~ = {200,150},
6102 <ugm>      {=} = {200,200},
6103 <m-t|bch|ebg|pmn|ppl|ptm|ugm> ( = {200, }, ) = { ,200},
6104 <cmr>      ( = {300, }, ) = { ,70},
6105 <m-t|ebg|ppl|ptm|ugm> / = {100,200},
6106 <cmr>      / = {100,100},
6107 <bch>      / = { ,150},
6108 <pmn>      / = {100,150},
6109 <m-t>      - = {300,300},
6110 <bch|ebg>    - = {300,400},
6111 <pmn>      - = {200,300},
6112 <cmr>      - = {500,300},
6113 <ppl>      - = {300,500},
6114 <ptm>      - = {500,500},
6115 <ugm>      - = {400,700},
6116 <blg>      - = {0,300},
6117 <m-t|pmn>    \textendash = {200,200}, \textendash = {150,150},
6118 <bch>      \textendash = {200,300}, \textendash = {150,200},
6119 <cmr>      \textendash = {500,300}, \textendash = {400,170},
6120 <ebg|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
6121 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
6122 <blg>      \textquoteleft = {400,400}, \textquoteright = {400,400},
6123 <cmr>      \textquoteleft = {800,200}, \textquoteright = {800,-20},
6124 <ebg>      \textquoteleft = {800,200}, \textquoteright = {800,200},
6125 <ppl>      \textquoteleft = {700,400}, \textquoteright = {700,400},
6126 <ptm>      \textquoteleft = {800,500}, \textquoteright = {800,500},
6127 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
6128 <blg>      \textquotedblright = {300,300}
6129 <cmr>      \textquotedblleft = {540,100}, \textquotedblright = {500,100}
6130 <ebg>      \textquotedblleft = {700,200}, \textquotedblright = {700,200}
6131 <ppl>      \textquotedblleft = {500,300}, \textquotedblright = {500,300}
6132 <ptm>      \textquotedblleft = {700,400}, \textquotedblright = {700,400}
6133 <ugm>      \textquotedblleft = {600,200}, \textquotedblright = {600,200}
6134 }
6135
6136 <*cmr|pmn>
6137 \SetProtrusion
6138 <cmr> [ name = cmr-it-OT1,
6139 <pmn> [ name = pmnj-it-OT1,
6140 <cmr> [ load = cmr-it ]
6141 <pmn> [ load = pmnj-it ]
6142 <cmr> { encoding = {OT1,OT4},
6143 <pmn> { encoding = OT1,
6144 <cmr> family = cmr,

```

```

6145 <pmn>    family   = pmnj,
6146 <cmr>    shape    = it    }
6147 <pmn>    shape    = {it,sl} }
6148 {
6149 <cmr>    \AE = {100,    },
6150 <pmn>    \AE = {    ,-50},
6151 <cmr>    \OE = {100,    },
6152 <pmn>    \OE = {50,    }
6153 <*cmr>
6154 "00 = {200,150}, % \Gamma
6155 "01 = {150,100}, % \Delta
6156 "02 = {150, 50}, % \Theta
6157 "03 = {150, 50}, % \Lambda
6158 "04 = {100,100}, % \Xi
6159 "05 = {100,100}, % \Pi
6160 "06 = {100, 50}, % \Sigma
6161 "07 = {200,150}, % \Upsilon
6162 "08 = {150, 50}, % \Phi
6163 "09 = {150,100}, % \Psi
6164 "0A = { 50, 50} % \Omega
6165 </cmr>
6166 }
6167
6168 </cmr|pmn>
6169 \SetProtrusion
6170 <m-t> [ name      = Tl-it-default,
6171 <bch> [ name      = bch-it-Tl,
6172 <blg> [ name      = blg-it-Tl,
6173 <cmr> [ name      = cmr-it-Tl,
6174 <ebg> [ name      = EBGaramond-it-Tl,
6175 <pmn> [ name      = pmnj-it-Tl,
6176 <ppl> [ name      = ppl-it-Tl,
6177 <ptm> [ name      = ptm-it-Tl,
6178 <ugm> [ name      = ugm-it-Tl,
6179 <m-t> load      = OTl-it ]
6180 <bch> load      = bch-it ]
6181 <blg> load      = blg-Tl ]
6182 <cmr> load      = cmr-it ]
6183 <pmn> load      = pmnj-it ]
6184 <ebg> load      = EBGaramond-it ]
6185 <ppl> load      = ppl-it ]
6186 <ptm> load      = ptm-it ]
6187 <ugm> load      = ugm-it ]
6188 <m-t|bch|cmr|ebg|pmn|ppl> { encoding = {Tl,Ly1},
6189 <blg|ptm|ugm> { encoding = Tl,
6190 <bch>    family   = bch,
6191 <blg>    family   = blg,
6192 <cmr>    family   = cmr,
6193 <pmn>    family   = pmnj,
6194 <ebg>    family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOf},
6195 <ppl>    family   = {ppl,pplx,pplj},
6196 <ptm>    family   = {ptm,ptmx,ptmj},
6197 <ugm>    family   = ugm,
6198 <m-t|bch|ebg|pmn|ppl|ptm> shape    = {it,sl} }
6199 <blg|cmr|ugm> shape    = it    }
6200 {
6201 <m-t|bch|pmn>    _ = {    ,100},
6202 <blg>    _ = {0,300},
6203 <cmr|ugm>    _ = {100,200},
6204 <ebg|ppl|ptm>    _ = {100,100},
6205 <blg>    . = {400,600},
6206 <blg>    {,}= {300,500},
6207 <cmr>    \AE = {100,    },
6208 <pmn>    \AE = {    ,-50},
6209 <bch|pmn>    \OE = { 50,    },

```



```

6210 <cmr> \OE = {100, },
6211 <pmn> 031 = { , -100}, % ff1
6212 <cmr|ptm> 156 = {100, }, % IJ
6213 <ebg> 156 = {50, }, % IJ
6214 <pmn> 156 = {20, }, % IJ
6215 <pmn> 188 = { , -30}, % ij
6216 <pmn> \v t = { , 100},
6217 <m-t|ebg|ppl|ptm> \textbackslash = {100,200},
6218 <cmr|ugm> \textbackslash = {300,300},
6219 <bch> \textbackslash = {150,150},
6220 <pmn> \textbackslash = {100,150},
6221 <ugm> \textbar = {200,200},
6222 <cmr> \textquotedblleft = {500,300},
6223 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
6224 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
6225 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
6226 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
6227 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6228 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6229 <ebg|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
6230 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
6231 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
6232 <bch|pmn> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
6233 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
6234 <ebg> \guilsinglleft = {500,400}, \guilsinglright = {300,500},
6235 <ugm> \guilsinglleft = {400,400}, \guilsinglright = {300,600},
6236 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
6237 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6238 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6239 <ebg> \guillemotleft = {300,300}, \guillemotright = {200,400},
6240 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
6241 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
6242 <m-t|ebg|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
6243 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
6244 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
6245 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
6246 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { , 200},
6247 <cmr|ebg|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
6248 <bch|pmn> \textless = {100, }, \textgreater = { , 100},
6249 <cmr|ebg|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
6250 <pmn> \textvisiblespace = {100,100}
6251 }
6252
6253 <*m-t|cmr|pmn>
6254 \SetProtrusion
6255 <m-t> [ name = T2A-it-default,
6256 <cmr> [ name = cmr-it-T2A,
6257 <pmn> [ name = pmnj-it-T2A,
6258 <m-t> load = OT1-it ]
6259 <cmr> load = cmr-it ]
6260 <pmn> load = pmnj-it ]
6261 { encoding = T2A,
6262 <cmr> family = cmr,
6263 <pmn> family = pmnj,
6264 <m-t|pmn> shape = {it,sl} }
6265 <cmr> shape = it }
6266 {
6267 <cmr> \CYRA = {100,50},
6268 <pmn> \CYRA = {50, },
6269 <cmr> \CYRB = {50, },
6270 <cmr> \CYRV = {50, },
6271 <pmn> \CYRV = {20,-50},
6272 <cmr> \CYRG = {100, },
6273 <pmn> \CYRG = {10, },
6274 <cmr> \CYRD = {50, },

```

```

6275 <cmr> \CYRE = {50, },
6276 <pmn> \CYRE = {20,-50},
6277 <cmr> \CYRZH = {50, },
6278 <cmr> \CYRZ = {50, },
6279 <pmn> \CYRZ = {20,-50},
6280 <cmr> \CYRI = {50, },
6281 <pmn> \CYRI = { , -30},
6282 <cmr> \CYRISHRT = {50, },
6283 <cmr> \CYRK = {50, },
6284 <pmn> \CYRK = {20, },
6285 <cmr> \CYRL = {50, },
6286 <cmr> \CYRM = {50, },
6287 <pmn> \CYRM = { , -30},
6288 <cmr> \CYRN = {50, },
6289 <cmr> \CYRO = {100, },
6290 <pmn> \CYRO = {50, },
6291 <cmr> \CYRP = {50, },
6292 <cmr> \CYRR = {50, },
6293 <pmn> \CYRR = {20,-50},
6294 <cmr> \CYRS = {100, },
6295 <pmn> \CYRS = {50, },
6296 <cmr> \CYRT = {100, },
6297 <pmn> \CYRT = {70, },
6298 <cmr> \CYRU = {100, },
6299 <pmn> \CYRU = {50, },
6300 <cmr> \CYRF = {100, },
6301 <cmr> \CYRH = {50, },
6302 <cmr> \CYRC = {50, },
6303 <cmr> \CYRCH = {100, },
6304 <cmr> \CYRSH = {50, },
6305 <cmr> \CYRSHCH = {50, },
6306 <cmr> \CYRHRDSN = {100, },
6307 <cmr> \CYRERY = {50, },
6308 <cmr> \CYRSFTSN = {50, },
6309 <cmr> \CYREREV = {50, },
6310 <cmr> \CYRYU = {50, },
6311 <cmr> \CYRYA = {50, },
6312 <pmn> \CYRYA = { , 20},
6313 <pmn> \cyrr = {-50, },
6314 <m-t|pmn> _ = { , 100},
6315 <cmr> _ = {100,200},
6316 <pmn> 031 = { , -100}, % ff1
6317 <pmn> \v t = { , 100},
6318 <m-t> \textbackslash = {100,200}, \quotedblbase = {400,500},
6319 <cmr> \textbackslash = {300,300}, \quotedblbase = {200,600},
6320 <pmn> \textbackslash = {100,150}, \quotedblbase = {150,500},
6321 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
6322 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6323 <pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6324 <m-t> \textbraceleft = {200,100}, \textbraceright = {200,200},
6325 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
6326 <pmn> \textbraceleft = {200, }, \textbraceright = { , 200},
6327 <cmr> \textquotedblleft = {500,300},
6328 <cmr> \textless = {300,100}, \textgreater = {200,100}
6329 <pmn> \textless = {100, }, \textgreater = { , 100}
6330 }
6331
6332 </m-t|cmr|pmn>
6333 <*m-t|ptm>
6334 \SetProtrusion
6335 <m-t> [ name = QX-it-default,
6336 <ptm> [ name = ptm-it-QX,
6337 <m-t> load = OT1-it ]
6338 <ptm> load = ptm-it ]
6339 { encoding = {QX},

```

```

6340 <ptm>    family    = {ptm,ptmx,ptmj},
6341    shape      = {it,sl} }
6342    {
6343 <ptm>      009 = { , 50}, % fk
6344    {=} = {100,100},
6345 <m-t>      \textunderscore = {100,100},
6346 <ptm>      \textunderscore = {100,150},
6347    \textbackslash = {100,200},
6348    \quotedblbase = {300,400},
6349 <m-t>      \guillemotleft = {300,300}, \guillemotright = {300,300},
6350 <ptm>      \guillemotleft = {200,400}, \guillemotright = {200,400},
6351    \textexclamdown = {200, }, \textquestiondown = {200, },
6352    \textbraceleft = {200,100}, \textbraceright = {200,200},
6353    \textless = {100,100}, \textgreater = {100,100},
6354    \textminus = {200,200}, \textdegree = {300,150},
6355 <m-t>      \copyright = {100,100}, \textregistered = {100,100}
6356 <ptm>      \textregistered = {100,150}, \copyright = {100,150},
6357 <ptm>      \textDelta = { 70, }, \textdelta = { , 50},
6358 <ptm>      \textpi = { 50, 80}, \textmu = { , 80},
6359 <ptm>      \texteuro = {200, }, \textellipsis = {100,200},
6360 <ptm>      \textquoteleft = {500,400}, \textquoteright = {500,400},
6361 <ptm>      \textquotedblleft = {500,300}, \textquotedblright = {400,400},
6362 <ptm>      \textapprox = { 50, 50}, \textinfty = {100,100},
6363 <ptm>      \textdagger = {150,150}, \textdaggerdbl = {100,100},
6364 <ptm>      \textdiv = {150,150}, \textasciitilde = { 80, 80},
6365 <ptm>      \texttimes = {100,150}, \textpm = { 50, 80},
6366 <ptm>      \textbullet = {300,100}, \textperiodcentered = {300,300},
6367 <ptm>      \textquotesingle = {500,500}, \textquotedbl = {300,300},
6368 <ptm>      \textperthousand = { ,50}
6369    }
6370
6371 </m-t|ptm>
6372 <*cmr|bch>
6373 \SetProtrusion
6374 <cmr> [ name = cmr-it-T5,
6375 <cmr>   load = cmr-it ]
6376 <bch> [ name = bch-it-T5,
6377 <bch>   load = bch-it ]
6378    { encoding = T5,
6379 <bch>      family    = bch,
6380 <cmr>      family    = cmr,
6381    shape      = it }
6382    {
6383 <bch>      _ = { ,100},
6384 <cmr>      _ = {100,200},
6385 <bch>      \textbackslash = {150,150},
6386 <cmr>      \textbackslash = {300,300},
6387 <bch>      \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6388 <cmr>      \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6389 <bch>      \guilsinglleft = {300,400}, \guilsinglright = {200,500},
6390 <cmr>      \guilsinglleft = {500,300}, \guilsinglright = {400,400},
6391 <bch>      \guillemotleft = {200,300}, \guillemotright = {150,400},
6392 <cmr>      \guillemotleft = {400,100}, \guillemotright = {200,300},
6393 <bch>      \textbraceleft = {200, }, \textbraceright = { ,200},
6394 <cmr>      \textbraceleft = {400,100}, \textbraceright = {200,200},
6395 <bch>      \textless = {100, }, \textgreater = { ,100},
6396 <cmr>      \textless = {300,100}, \textgreater = {200,100}
6397    }
6398
6399 </cmr|bch>

```

Slanted is very similar to italic.

```

6400 <*cmr>
6401 \SetProtrusion
6402 [ name = cmr-sl,

```

```

6403     load      = cmr-it-OT1 ]
6404     { encoding = {OT1,OT4},
6405       family   = cmr,
6406       shape    = sl  }
6407     {
6408       L = { ,50},
6409       f = { ,-50},
6410       - = {300, },
6411       \textendash = {400, }, \textemdash = {300, }
6412     }
6413
6414 \SetProtrusion
6415 [ name      = cmr-sl-T1,
6416   load      = cmr-it-T1 ]
6417 { encoding = {T1,LY1},
6418   family   = cmr,
6419   shape    = sl  }
6420 {
6421   L = { ,50},
6422   f = { ,-50},
6423   - = {300, },
6424   \textendash = {400, }, \textemdash = {300, }
6425 }
6426
6427 \SetProtrusion
6428 [ name      = cmr-sl-T2A,
6429   load      = cmr-it-T2A ]
6430 { encoding = T2A,
6431   family   = cmr,
6432   shape    = sl  }
6433 {
6434   L = { ,50},
6435   f = { ,-50},
6436   - = {300, },
6437   \textendash = {400, }, \textemdash = {300, }
6438 }
6439
6440 \SetProtrusion
6441 [ name      = cmr-sl-T5,
6442   load      = cmr-it-T5 ]
6443 { encoding = T5,
6444   family   = cmr,
6445   shape    = sl  }
6446 {
6447   L = { ,50},
6448   f = { ,-50},
6449   - = {300, },
6450   \textendash = {400, }, \textemdash = {300, }
6451 }
6452
6453 \SetProtrusion
6454 [ name      = lmr-it-T1,
6455   load      = cmr-it-T1 ]
6456 { encoding = {T1,LY1},
6457   family   = lmr,
6458   shape    = {it,sl} }
6459 {
6460   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
6461   \quotesinglbase    = { ,400}, \quotedblbase      = { ,500}
6462 }
6463

```

Oldstyle numerals are slightly different.

```

6464 \SetProtrusion
6465 [ name = cmr(oldstyle)-it,

```

```

6466     load = cmr-it-T1 ]
6467     { encoding = T1,
6468       family   = {hfor,cmor},
6469       shape    = {it,sl} }
6470     {
6471       1 = {250, 50},
6472       2 = {150,-100},
6473       3 = {100,-50},
6474       4 = {150,150},
6475       6 = {200,  },
6476       7 = {200, 50},
6477       8 = {150,-50},
6478       9 = {100, 50}
6479     }
6480
6481 </cmr>
6482 <*pmn>
6483 \SetProtrusion
6484 [ name   = pmnx-it,
6485   load   = pmnj-it ]
6486 { encoding = OT1,
6487   family   = pmnx,
6488   shape    = {it,sl} }
6489 {
6490   1 = {100,150}
6491 }
6492
6493 \SetProtrusion
6494 [ name   = pmnx-it-T1,
6495   load   = pmnj-it-T1 ]
6496 { encoding = {T1,LY1},
6497   family   = pmnx,
6498   shape    = {it,sl} }
6499 {
6500   1 = {100,150}
6501 }
6502
6503 \SetProtrusion
6504 [ name   = pmnx-it-T2A,
6505   load   = pmnj-it-T2A ]
6506 { encoding = {T2A},
6507   family   = pmnx,
6508   shape    = {it,sl} }
6509 {
6510   1 = {100,150}
6511 }
6512
6513 </pmn>
6514 <*ptm>
6515 \SetProtrusion
6516 [ name   = ptm-it-LY1,
6517   load   = ptm-it-T1 ]
6518 { encoding = {LY1},
6519   family   = {ptm,ptmx,ptmj},
6520   shape    = {it,sl} }
6521 {
6522   - = {100,100},
6523   \texttrademark = {100,100},
6524   \textregistered = {100,100},
6525   \textcopyright = {100,100},
6526   \textdegree = {300,100},
6527   \textminus = {200,200},
6528   \textellipsis = {100,200},
6529 % \texteuro = { , }, % ?
6530   \textcent = {100,100},

```

```

6531 \textquotesingle = {500, },
6532 \textflorin = {100, 70},
6533 \textdagger = {150,150},
6534 \textdaggerdbl = {100,100},
6535 \textbullet = {150,150},
6536 \textonesuperior = {150,100},
6537 \texttwosuperior = {150, 50},
6538 \textthreesuperior = {150, 50},
6539 \textparagraph = {100, },
6540 \textperiodcentered = {500,300},
6541 \textonequarter = { 50, },
6542 \textonehalf = { 50, },
6543 \textplusminus = {100,100},
6544 \textmultiply = {150,150},
6545 \textdivide = {150,150}
6546 }
6547
6548 </ptm>

```

### 15.8.3 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).

```

6549 <*(big|ugm)>
6550 \SetProtrusion
6551 <m-t> [ name = OT1-sc,
6552 <bch> [ name = bch-sc,
6553 <cmr> [ name = cmr-sc-OT1,
6554 <ebg> [ name = EBGaramond-sc,
6555 <pmn> [ name = pmnj-sc,
6556 <ppl> [ name = ppl-sc,
6557 <ptm> [ name = ptm-sc,
6558 <m-t> load = default ]
6559 <bch> load = bch-default ]
6560 <cmr> load = cmr-OT1 ]
6561 <ebg> load = EBGaramond-default ]
6562 <pmn> load = pmnj-default ]
6563 <ppl> load = ppl-default ]
6564 <ptm> load = ptm-default ]
6565 <m-t|bch|ebg|pmn> { encoding = OT1,
6566 <cmr|ppl|ptm> { encoding = {OT1,OT4},
6567 <bch> family = bch,
6568 <cmr> family = cmr,
6569 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOf},
6570 <pmn> family = pmnj,
6571 <ppl> family = {ppl,pplx,pplj},
6572 <ptm> family = {ptm,ptmx,ptmj},
6573 shape = sc }
6574 {
6575 a = {50,50},
6576 <cmr|ebg|ppl|ptm> \ae = {50, },
6577 <bch|pmn> c = {50, },
6578 <bch|ebg|pmn> d = { ,50},
6579 <m-t|bch|cmr|ebg|pmn|ptm> f = { ,50},
6580 <bch|ebg|pmn> g = {50, },
6581 <m-t|cmr|ebg|pmn|ppl|ptm> j = {50, },
6582 <bch> j = {100, },
6583 <m-t|bch|cmr|ebg|pmn|ppl> l = { ,50},
6584 <ptm> l = { ,80},
6585 <m-t|bch|cmr|ebg|pmn|ppl> 013 = { ,50}, % fl
6586 <ptm> 013 = { ,80}, % fl
6587 <bch|ebg|pmn> o = {50,50},

```

```

6588 <ebg|pmn> \oe = {50, },
6589 <ppl> p = { 0, 0},
6590 <bch|ebg|pmn> q = {50,70},
6591 <ppl> q = { 0, },
6592 <m-t|cmr|ebg|pmn|ppl|ptm> r = { , 0},
6593 t = {50,50},
6594 <m-t|bch|cmr|ebg|pmn|ppl> y = {50,50}
6595 <ptm> y = {80,80}
6596 }
6597
6598 \SetProtrusion
6599 <m-t> [ name = T1-sc,
6600 <bch> [ name = bch-sc-T1,
6601 <cmr> [ name = cmr-sc-T1,
6602 <ebg> [ name = EBGaramond-sc-T1,
6603 <pmn> [ name = pmnj-sc-T1,
6604 <ppl> [ name = ppl-sc-T1,
6605 <ptm> [ name = ptm-sc-T1,
6606 <m-t> load = T1-default ]
6607 <bch> load = bch-T1 ]
6608 <cmr> load = cmr-T1 ]
6609 <ebg> load = EBGaramond-T1 ]
6610 <pmn> load = pmnj-T1 ]
6611 <ppl> load = ppl-T1 ]
6612 <ptm> load = ptm-T1 ]
6613 { encoding = {T1,LY1},
6614 <bch> family = bch,
6615 <cmr> family = cmr,
6616 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-Of,EBGaramond-TOsF},
6617 <pmn> family = pmnj,
6618 <ppl> family = {ppl,pplx,pplj},
6619 <ptm> family = {ptm,ptmx,ptmj},
6620 shape = sc }
6621 {
6622 a = {50,50},
6623 <cmr|ebg|ppl|ptm> \ae = {50, },
6624 <bch|pmn> c = {50, },
6625 <bch|ebg|pmn> d = { ,50},
6626 <m-t|bch|cmr|ebg|pmn|ptm> f = { ,50},
6627 <bch|ebg|pmn> g = {50, },
6628 <m-t|cmr|ebg|pmn|ppl|ptm> j = {50, },
6629 <bch> j = {100, },
6630 <m-t|bch|cmr|ebg|pmn|ppl> l = { ,50},
6631 <ptm> l = { ,80},
6632 <m-t|bch|cmr|ebg|pmn|ppl> 029 = { ,50}, % fl
6633 <ptm> 029 = { ,80}, % fl
6634 <bch|ebg|pmn> o = {50,50},
6635 <bch|ebg|pmn> \oe = {50, },
6636 <ppl> p = { 0, 0},
6637 <bch|ebg|pmn> q = {50,70},
6638 <ppl> q = { 0, },
6639 <m-t|cmr|ebg|pmn|ppl|ptm> r = { , 0},
6640 t = {50,50},
6641 <m-t|bch|cmr|ebg|pmn|ppl> y = {50,50}
6642 <ptm> y = {80,80}
6643 }
6644
6645 </(blg|ugm)>
6646 <*m-t|cmr>
6647 \SetProtrusion
6648 <m-t> [ name = T2A-sc,
6649 <cmr> [ name = cmr-sc-T2A,
6650 <m-t> load = T2A-default ]
6651 <cmr> load = cmr-T2A ]
6652 { encoding = T2A,

```

```

6653 <cmr>    family    = cmr,
6654    shape      = sc }
6655    {
6656    \cyra = {50,50},
6657    \cyrg = { ,50},
6658    \cyrt = {50,50},
6659    \cyry = { ,50}
6660    }
6661
6662 </m-t|cmr>
6663 <*m-t>
6664 \SetProtrusion
6665 [ name      = QX-sc,
6666   load      = QX-default ]
6667 { encoding = QX,
6668   shape     = sc }
6669 {
6670   a = {50,50},
6671   f = { ,50},
6672   j = {50, },
6673   l = { ,50},
6674   013 = { ,50}, % fl
6675   r = { , 0},
6676   t = {50,50},
6677   y = {50,50}
6678 }
6679
6680 </m-t>
6681 <*cmr|bch>
6682 \SetProtrusion
6683 <bch> [ name      = bch-sc-T5,
6684 <bch>   load      = bch-T5 ]
6685 <cmr> [ name      = cmr-sc-T5,
6686 <cmr>   load      = cmr-T5 ]
6687 { encoding = T5,
6688 <bch>   family    = bch,
6689 <cmr>   family    = cmr,
6690   shape     = sc }
6691 {
6692   a = {50,50},
6693 <bch>   c = {50, },
6694 <bch>   d = { ,50},
6695   f = { ,50},
6696 <bch>   g = {50, },
6697 <bch>   j = {100, },
6698 <cmr>   j = {50, },
6699   l = { ,50},
6700 <bch>   o = {50,50},
6701 <bch>   q = { 0, },
6702 <cmr>   r = { , 0},
6703   t = {50,50},
6704   y = {50,50}
6705 }
6706
6707 </cmr|bch>
6708 <*pmn>
6709 \SetProtrusion
6710 [ name      = pmnx-sc,
6711   load      = pmnj-sc ]
6712 { encoding = OT1,
6713   family    = pmnx,
6714   shape     = sc }
6715 {
6716   1 = {230,180}
6717 }

```



```

6718
6719 \SetProtrusion
6720 [ name      = pmnx-sc-T1,
6721   load      = pmnj-sc-T1 ]
6722 { encoding = {T1,LY1},
6723   family   = pmnx,
6724   shape     = sc }
6725 {
6726   1 = {230,180}
6727 }
6728

```

#### 15.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

6729 \SetProtrusion
6730 [ name      = pmnj-scit,
6731   load      = pmnj-it   ]
6732 { encoding = OT1,
6733   family   = pmnj,
6734   shape     = {scit,si} }
6735 {
6736   a = {50, },
6737   \ae = { , -50},
6738   b = {20, -50},
6739   c = {50, -50},
6740   d = {20, 0},
6741   e = {20, -50},
6742   f = {10, 0},
6743   012 = {10, -50}, % fi
6744   013 = {10, -50}, % fl
6745   014 = {10, -50}, % ffi
6746   015 = {10, -50}, % ffl
6747   g = {50, -50},
6748   i = {20, -50},
6749   j = {20, 0},
6750   k = {20, },
6751   l = {20, 50},
6752   m = { , -30},
6753   n = { , -30},
6754   o = {50, },
6755   \oe = {50, -50},
6756   p = {20, -50},
6757   q = {50, },
6758   r = {20, 0},
6759   s = {20, -30},
6760   t = {70, },
6761   u = {50, -50},
6762   v = {100, },
6763   w = {100, },
6764   y = {50, },
6765   z = { , -50}
6766 }
6767
6768 \SetProtrusion
6769 [ name      = pmnj-scit-T1,
6770   load      = pmnj-it-T1   ]
6771 { encoding = {T1,LY1},
6772   family   = pmnj,
6773   shape     = {scit,si}    }
6774 {
6775   a = {50, },
6776   \ae = { , -50},

```

```

6777     b = {20,-50},
6778     c = {50,-50},
6779     d = {20, 0},
6780     e = {20,-50},
6781     f = {10, 0},
6782     028 = {10,-50}, % fi
6783     029 = {10,-50}, % fl
6784     030 = {10,-50}, % ffi
6785     031 = {10,-50}, % ffl
6786     g = {50,-50},
6787     i = {20,-50},
6788     188 = {20, 0}, % ij
6789     j = {20, 0},
6790     k = {20, },
6791     l = {20,50},
6792     m = { , -30},
6793     n = { , -30},
6794     o = {50, },
6795     \oe = {50,-50},
6796     p = {20,-50},
6797     q = {50, },
6798     r = {20, 0},
6799     s = {20,-30},
6800     t = {70, },
6801     u = {50,-50},
6802     v = {100, },
6803     w = {100, },
6804     y = {50, },
6805     z = { , -50}
6806 }
6807
6808 \SetProtrusion
6809 [ name    = pmnx-scit,
6810   load    = pmnj-scit ]
6811 { encoding = OT1,
6812   family   = pmnx,
6813   shape    = {scit,si} }
6814 {
6815   1 = {100,150}
6816 }
6817
6818 \SetProtrusion
6819 [ name    = pmnx-scit-T1,
6820   load    = pmnj-scit-T1 ]
6821 { encoding = {T1,LY1},
6822   family   = pmnx,
6823   shape    = {scit,si} }
6824 {
6825   1 = {100,150}
6826 }
6827
6828 </pmn>

```

### 15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

6829 \SetProtrusion
6830 <m-t> [ name    = textcomp ]
6831 <bch> [ name    = bch-textcomp ]
6832 <blg> [ name    = blg-textcomp ]
6833 <cmr> [ name    = cmr-textcomp ]
6834 <ebg> [ name    = EBGaramond-textcomp ]
6835 <pmn> [ name    = pmn-textcomp ]

```

```

6836 <ppl> [ name = ppl-textcomp ]
6837 <ptm> [ name = ptm-textcomp ]
6838 <ugm> [ name = ugm-textcomp ]
6839 <m-t> { encoding = TS1 }
6840 <!m-t> { encoding = TS1,
6841 <bch> family = bch }
6842 <blg> family = blg }
6843 <cmr> family = cmr }
6844 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF} }
6845 <pmn> family = {pmnx,pmnj} }
6846 <ppl> family = {ppl,pplx,pplj} }
6847 <ptm> family = {ptm,ptmx,ptmj} }
6848 <ugm> family = ugm }
6849 {
6850 <blg> \textquotestraightbase = {400,500},
6851 <cmr> \textquotestraightbase = {300,300},
6852 <ebg|pmn> \textquotestraightbase = {400,400},
6853 <blg> \textquotestraightdblbase = {300,400},
6854 <cmr|pmn> \textquotestraightdblbase = {300,300},
6855 <ebg> \textquotestraightdblbase = {400,400},
6856 <bch|cmr|ebg|pmn|ugm> \texttwelvewdash = {200,200},
6857 <bch|cmr|ebg|pmn> \textthreequartersemdash = {150,150},
6858 <ugm> \textthreequartersemdash = {200,200},
6859 <blg> \textquotesingle = {500,600},
6860 <cmr|pmn> \textquotesingle = {300,400},
6861 <ebg> \textquotesingle = {400,500},
6862 <ptm> \textquotesingle = {500,500},
6863 <ugm> \textquotesingle = {300,500},
6864 <bch|cmr|pmn> \textasteriskcentered = {200,300},
6865 <blg> \textasteriskcentered = {150,200},
6866 <ebg> \textasteriskcentered = {300,300},
6867 <ugm> \textasteriskcentered = {100,200},
6868 <pmn> \textfrown = {-200,-200},
6869 <cmr> \textoneoldstyle = {100,100},
6870 <pmn> \textoneoldstyle = { , 50},
6871 <cmr> \textthreeoldstyle = { , 50},
6872 <ebg|pmn> \textthreeoldstyle = { 50, },
6873 <cmr> \textfouroldstyle = { 50, 50},
6874 <ebg|pmn> \textfouroldstyle = { 50, },
6875 <cmr|ebg|pmn> \textsevenoldstyle = { 50, 80},
6876 <cmr> \textlangle = {400, },
6877 <cmr> \textrangle = { , 400},
6878 <m-t|bch|pmn|ptm> \textminus = {200,200},
6879 <cmr|ebg|ppl> \textminus = {300,300},
6880 <blg|ugm> \textminus = {250,300},
6881 <bch|ebg|pmn> \textlbrackdbl = {100, },
6882 <blg> \textlbrackdbl = {200, },
6883 <bch|ebg|pmn> \textrbrackdbl = { , 100},
6884 <blg> \textrbrackdbl = { , 200},
6885 <pmn> \textasciigrave = {200,500},
6886 <bch|blg|cmr|ebg|pmn> \texttildebelow = {200,250},
6887 <pmn> \textasciibreve = {300,400},
6888 <pmn> \textasciicaron = {300,400},
6889 <pmn> \textacutetilde = {200,300},
6890 <pmn> \textgravedbl = {150,300},
6891 <bch|pmn|ugm> \textdagger = { 80, 80},
6892 <blg> \textdagger = {200,200},
6893 <cmr|ebg> \textdagger = {100,100},
6894 <ptm> \textdagger = {150,150},
6895 <blg> \textdaggerdbl = {150,150},
6896 <cmr|ebg|pmn> \textdaggerdbl = { 80, 80},
6897 <ptm> \textdaggerdbl = {100,100},
6898 <bch> \textbardbl = {100,100},
6899 <blg|ugm> \textbardbl = {150,150},
6900 <bch> \textbullet = {200,200},

```

```

6901 <blg> \textbullet = {400,500},
6902 <cmr|ebg|pmn> \textbullet = { ,100},
6903 <ptm> \textbullet = {150,150},
6904 <ugm> \textbullet = { 50,100},
6905 <bch|cmr|pmn> \textcelsius = { 50, },
6906 <ebg> \textcelsius = { 80, },
6907 <bch> \textflorin = { 50, 50},
6908 <blg> \textflorin = {100,100},
6909 <ebg|ugm> \textflorin = { ,100},
6910 <pmn> \textflorin = { 50,100},
6911 <ptm> \textflorin = { 50, 70},
6912 <cmr> \textcolonmonetary = { , 50},
6913 <ebg|pmn> \textcolonmonetary = { 50, },
6914 <pmn> \textinterrobang = { ,100},
6915 <pmn> \textinterrobangdown = {100, },
6916 <m-t|ebg|ptm> \texttrademark = {100,100},
6917 <bch> \texttrademark = {150,150},
6918 <blg|cmr|ppl> \texttrademark = {200,200},
6919 <pmn> \texttrademark = { 50, 50},
6920 <ugm> \texttrademark = {100,150},
6921 <bch|ugm> \textcent = { 50, },
6922 <ptm> \textcent = {100,100},
6923 <bch> \textsterling = { 50, },
6924 <ugm> \textsterling = { , 50},
6925 <bch> \textbrokenbar = {200,200},
6926 <blg> \textbrokenbar = {250,250},
6927 <ugm> \textbrokenbar = {200,300},
6928 <pmn> \textasciidieresis = {300,400},
6929 <m-t|bch|cmr|ebg|ptm|ugm> \textcopyright = {100,100},
6930 <pmn> \textcopyright = {100,150},
6931 <ppl> \textcopyright = {200,200},
6932 <bch|cmr|ugm> \textordfeminine = {100,200},
6933 <ebg|pmn> \textordfeminine = {200,200},
6934 <bch|cmr|ebg|pmn|ugm> \textlnot = {200, },
6935 <blg> \textlnot = {200,100},
6936 <m-t|bch|cmr|ebg|ptm|ugm> \textregistered = {100,100},
6937 <pmn> \textregistered = { 50,150},
6938 <ppl> \textregistered = {200,200},
6939 <pmn> \textasciimacron = {150,200},
6940 <m-t|ppl|ptm> \textdegree = {300,300},
6941 <bch> \textdegree = {150,200},
6942 <blg|ugm> \textdegree = {200,200},
6943 <cmr|ebg> \textdegree = {400,400},
6944 <pmn> \textdegree = {150,400},
6945 <bch|cmr|ebg|pmn|ugm> \textpm = {150,200},
6946 <blg> \textpm = {100,100},
6947 <ptm> \textpm = { 50, 80},
6948 <bch|blg|ugm> \texttwosuperior = {100,200},
6949 <cmr> \texttwosuperior = { 50,100},
6950 <ebg|pmn> \texttwosuperior = {200,200},
6951 <ptm> \texttwosuperior = { 50, 50},
6952 <bch|blg|ugm> \textthreesuperior = {100,200},
6953 <cmr> \textthreesuperior = { 50,100},
6954 <ebg|pmn> \textthreesuperior = {200,200},
6955 <ptm> \textthreesuperior = { 50, 50},
6956 <pmn> \textasciiacute = {300,400},
6957 <bch|ugm> \textmu = { ,100},
6958 <bch|ebg|pmn> \textparagraph = { ,100},
6959 <bch|cmr|ebg|pmn> \textperiodcentered = {300,400},
6960 <blg> \textperiodcentered = {400,500},
6961 <ptm> \textperiodcentered = {300,300},
6962 <ugm> \textperiodcentered = {200,500},
6963 <bch|blg|ugm> \textonesuperior = {200,300},
6964 <cmr|ebg|pmn> \textonesuperior = {200,200},
6965 <ptm> \textonesuperior = {100,100},

```

```

6966 <bch|ebg|pmn|ugm> \textordmasculine = {200,200},
6967 <blg|cmr> \textordmasculine = {100,200},
6968 <bch|cmr|pmn> \texteuro = {100, },
6969 <ebg> \texteuro = { 50,100},
6970 <bch> \texttimes = {200,200},
6971 <blg|ptm> \texttimes = {100,100},
6972 <cmr> \texttimes = {150,250},
6973 <ebg> \texttimes = {100,150},
6974 <pmn> \texttimes = { 70,100},
6975 <ugm> \texttimes = {200,300},
6976 <bch|ebg|pmn> \textdiv = {150,200}
6977 <blg> \textdiv = {100,100}
6978 <cmr> \textdiv = {150,250}
6979 <ptm> \textdiv = { 50,100},
6980 <ugm> \textdiv = {200,300},
6981 <ptm> \textperthousand = { ,50}
6982 <ugm> \textsection = { ,100},
6983 <ugm> \textonehalf = { 50,100},
6984 <ugm> \textonequarter = { 50,100},
6985 <ugm> \textthreequarters = { 50,100},
6986 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

6987 }
6988
6989 <*cmr|ebg|pmn|ugm>
6990 \SetProtrusion
6991 <cmr> [ name = cmr-textcomp-it ]
6992 <ebg> [ name = EBGaramond-textcomp-it ]
6993 <pmn> [ name = pmn-textcomp-it ]
6994 <ugm> [ name = ugm-textcomp-it ]
6995 { encoding = TS1,
6996 <cmr> family = cmr,
6997 <ebg> family = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OfF,EBGaramond-TOfF},
6998 <pmn> family = {pmnx,pmnj},
6999 <ugm> family = ugm,
7000 <!ugm> shape = {it,sl} }
7001 <ugm> shape = it }
7002 {
7003 <cmr> \textquotestraightbase = {300,600},
7004 <ebg|pmn> \textquotestraightbase = {400,400},
7005 <cmr> \textquotestraightdblbase = {300,600},
7006 <ebg> \textquotestraightdblbase = {300,400},
7007 <pmn> \textquotestraightdblbase = {300,300},
7008 \texttwelvewardash = {200,200},
7009 <cmr|ebg|pmn> \textthreequartersemdash = {150,150},
7010 <ugm> \textthreequartersemdash = {200,200},
7011 <cmr> \textquotesingle = {600,300},
7012 <ebg> \textquotesingle = {800,100},
7013 <pmn> \textquotesingle = {300,200},
7014 <ugm> \textquotesingle = {500,500},
7015 <cmr> \textasteriskcentered = {300,200},
7016 <ebg> \textasteriskcentered = {500,100},
7017 <pmn> \textasteriskcentered = {200,300},
7018 <ugm> \textasteriskcentered = {300,150},
7019 <pmn> \textfractionsolidus = {-200,-200},
7020 <cmr> \textoneoldstyle = {100, 50},
7021 <ebg> \textoneoldstyle = {100, },
7022 <pmn> \textoneoldstyle = { 50, },
7023 <ebg> \texttwooldstyle = { 50, },
7024 <pmn> \texttwooldstyle = {-50, },
7025 <cmr> \textthreeoldstyle = {100, 50},
7026 <pmn> \textthreeoldstyle = {-100, },
7027 <cmr> \textfouroldstyle = { 50, 50},
7028 <ebg> \textfouroldstyle = { 50,100},

```

7029	<i>cmr</i>	<code>\textsevenoldstyle</code>	= { 50, 80},
7030	<i>ebg</i>	<code>\textsevenoldstyle</code>	= { 50, },
7031	<i>pmn</i>	<code>\textsevenoldstyle</code>	= { 20, },
7032	<i>cmr</i>	<code>\textlangle</code>	= {400, },
7033	<i>cmr</i>	<code>\textrangle</code>	= { ,400},
7034	<i>cmr ebg</i>	<code>\textminus</code>	= {300,300},
7035	<i>pmn</i>	<code>\textminus</code>	= {200,200},
7036	<i>ugm</i>	<code>\textminus</code>	= {250,300},
7037	<i>ebg pmn</i>	<code>\textlbrackdbl</code>	= {100, },
7038	<i>ebg pmn</i>	<code>\textrbrackdbl</code>	= { ,100},
7039	<i>pmn</i>	<code>\textasciigrave</code>	= {300,300},
7040	<i>cmr ebg pmn</i>	<code>\texttildelow</code>	= {200,250},
7041	<i>pmn</i>	<code>\textasciibreve</code>	= {300,300},
7042	<i>pmn</i>	<code>\textasciicaron</code>	= {300,300},
7043	<i>pmn</i>	<code>\textacutedbl</code>	= {200,300},
7044	<i>pmn</i>	<code>\textgravedbl</code>	= {150,300},
7045	<i>cmr</i>	<code>\textdagger</code>	= {100,100},
7046	<i>ebg</i>	<code>\textdagger</code>	= {200,100},
7047	<i>pmn</i>	<code>\textdagger</code>	= { 80, 50},
7048	<i>ugm</i>	<code>\textdagger</code>	= { 80, 80},
7049	<i>cmr ebg</i>	<code>\textdaggerdbl</code>	= { 80, 80},
7050	<i>pmn</i>	<code>\textdaggerdbl</code>	= { 80, 50},
7051	<i>ugm</i>	<code>\textbardbl</code>	= {150,150},
7052	<i>cmr</i>	<code>\textbullet</code>	= {200,100},
7053	<i>ebg</i>	<code>\textbullet</code>	= {300, },
7054	<i>pmn</i>	<code>\textbullet</code>	= { 30, 70},
7055	<i>ugm</i>	<code>\textbullet</code>	= { 50,100},
7056	<i>cmr</i>	<code>\textcelsius</code>	= {100, },
7057	<i>ebg</i>	<code>\textcelsius</code>	= {200, },
7058	<i>pmn</i>	<code>\textcelsius</code>	= { 50,-50},
7059	<i>ebg</i>	<code>\textflorin</code>	= {100, },
7060	<i>pmn</i>	<code>\textflorin</code>	= { 50,100},
7061	<i>ugm</i>	<code>\textflorin</code>	= { ,100},
7062	<i>cmr</i>	<code>\textcolonmonetary</code>	= {150, },
7063	<i>ebg</i>	<code>\textcolonmonetary</code>	= {100, },
7064	<i>pmn</i>	<code>\textcolonmonetary</code>	= { 50,-50},
7065	<i>cmr ebg</i>	<code>\texttrademark</code>	= {200, },
7066	<i>pmn</i>	<code>\texttrademark</code>	= { 50,100},
7067	<i>ugm</i>	<code>\texttrademark</code>	= {150, 50},
7068	<i>ugm</i>	<code>\textcent</code>	= { 50, },
7069	<i>ugm</i>	<code>\textsterling</code>	= { , 50},
7070	<i>ugm</i>	<code>\textbrokenbar</code>	= {200,300},
7071	<i>pmn</i>	<code>\textasciidieresis</code>	= {300,200},
7072	<i>cmr</i>	<code>\textcopyright</code>	= {100, },
7073	<i>ebg</i>	<code>\textcopyright</code>	= {200,100},
7074	<i>pmn</i>	<code>\textcopyright</code>	= {100,150},
7075	<i>ugm</i>	<code>\textcopyright</code>	= {300, },
7076	<i>cmr</i>	<code>\textordfeminine</code>	= {100,100},
7077	<i>pmn</i>	<code>\textordfeminine</code>	= {200,200},
7078	<i>ugm</i>	<code>\textordfeminine</code>	= {100,200},
7079	<i>cmr ebg</i>	<code>\textlnot</code>	= {300, },
7080	<i>pmn ugm</i>	<code>\textlnot</code>	= {200, },
7081	<i>cmr</i>	<code>\textregistered</code>	= {100, },
7082	<i>ebg</i>	<code>\textregistered</code>	= {200,100},
7083	<i>pmn</i>	<code>\textregistered</code>	= { 50,150},
7084	<i>ugm</i>	<code>\textregistered</code>	= {300, },
7085	<i>pmn</i>	<code>\textasciimacron</code>	= {150,200},
7086	<i>cmr ebg</i>	<code>\textdegree</code>	= {500,100},
7087	<i>pmn</i>	<code>\textdegree</code>	= {150,150},
7088	<i>ugm</i>	<code>\textdegree</code>	= {300,200},
7089	<i>cmr</i>	<code>\textpm</code>	= {150,100},
7090	<i>ebg</i>	<code>\textpm</code>	= {200,150},
7091	<i>pmn ugm</i>	<code>\textpm</code>	= {150,200},
7092	<i>cmr</i>	<code>\textonesuperior</code>	= {400, },
7093	<i>ebg</i>	<code>\textonesuperior</code>	= {300,100},

```

7094 <pmn> \textonesuperior = {200,100},
7095 <ugm> \textonesuperior = {300,300},
7096 <cmr> \texttwosuperior = {400, },
7097 <ebg> \texttwosuperior = {300, },
7098 <pmn> \texttwosuperior = {200,100},
7099 <ugm> \texttwosuperior = {300,200},
7100 <cmr> \textthreesuperior = {400, },
7101 <ebg> \textthreesuperior = {300, },
7102 <pmn> \textthreesuperior = {200,100},
7103 <ugm> \textthreesuperior = {300,200},
7104 <ugm> \textmu = { ,100},
7105 <pmn> \textasciicute = {300,200},
7106 <cmr> \textparagraph = {200, },
7107 <pmn> \textparagraph = { ,100},
7108 <cmr> \textperiodcentered = {500,500},
7109 <ebg|pmn|ugm> \textperiodcentered = {300,400},
7110 <cmr> \textordmasculine = {100,100},
7111 <pmn> \textordmasculine = {200,200},
7112 <ugm> \textordmasculine = {300,200},
7113 <cmr> \texteuro = {200, },
7114 <ebg> \texteuro = {100, },
7115 <pmn> \texteuro = {100,-50},
7116 <cmr> \texttimes = {200,200},
7117 <ebg> \texttimes = {200,100},
7118 <pmn> \texttimes = { 70,100},
7119 <ugm> \texttimes = {200,300},
7120 <cmr|ebg> \textdiv = {200,200},
7121 <pmn> \textdiv = {150,200},
7122 <ugm> \textdiv = {200,300},
7123 <ugm> \textsection = { ,200},
7124 <ugm> \textonehalf = { 50,100},
7125 <ugm> \textonequarter = { 50,100},
7126 <ugm> \textthreequarters = { 50,100},
7127 <ugm> \textsurd = { ,100}
7128 }
7129
7130 </cmr|ebg|pmn|ugm>

```

### 15.8.6 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}

```

```

7131 <*cmr>
7132 \SetProtrusion
7133 [ name = cmr-math-letters ]
7134 { encoding = OML,
7135   family = cmm,
7136   series = {m,b},
7137   shape = it }

```

```

7138 {
7139     A = {100, 50}, % \mathnormal
7140     B = { 50,   },
7141     C = { 50,   },
7142     D = { 50, 50},
7143     E = { 50,   },
7144     F = {100, 50},
7145     G = { 50, 50},
7146     H = { 50, 50},
7147     I = { 50, 50},
7148     J = {150, 50},
7149     K = { 50,100},
7150     L = { 50, 50},
7151     M = { 50,   },
7152     N = { 50,   },
7153     O = { 50,   },
7154     P = { 50,   },
7155     Q = { 50, 50},
7156     R = { 50,   },
7157     S = { 50,   },
7158     T = { 50,100},
7159     U = { 50, 50},
7160     V = {100,100},
7161     W = { 50,100},
7162     X = { 50,100},
7163     Y = {100,100},
7164     f = {100,100},
7165     h = {   ,100},
7166     i = {   , 50},
7167     j = {   , 50},
7168     k = {   , 50},
7169     r = {   , 50},
7170     v = {   , 50},
7171     w = {   , 50},
7172     x = {   , 50},
7173     "0B = { 50,100}, % \alpha
7174     "0C = { 50, 50}, % \beta
7175     "0D = {200,150}, % \gamma
7176     "0E = { 50, 50}, % \delta
7177     "0F = { 50, 50}, % \epsilon
7178     "10 = { 50,150}, % \zeta
7179     "12 = { 50,   }, % \theta
7180     "13 = {   ,100}, % \iota
7181     "14 = {   ,100}, % \kappa
7182     "15 = {100, 50}, % \lambda
7183     "16 = {   , 50}, % \mu
7184     "17 = {   , 50}, % \nu
7185     "18 = {   , 50}, % \xi
7186     "19 = { 50,100}, % \pi
7187     "1A = { 50, 50}, % \rho
7188     "1B = {   ,150}, % \sigma
7189     "1C = { 50,150}, % \tau
7190     "1D = { 50, 50}, % \upsilon
7191     "1F = { 50,100}, % \chi
7192     "20 = { 50, 50}, % \psi
7193     "21 = {   , 50}, % \omega
7194     "22 = {   , 50}, % \varepsilon
7195     "23 = {   , 50}, % \vartheta
7196     "24 = {   , 50}, % \varpi
7197     "25 = {100,   }, % \varrho
7198     "26 = {100,100}, % \varsigma
7199     "27 = { 50, 50}, % \varphi
7200     "28 = {100,100}, % \leftharpoonup
7201     "29 = {100,100}, % \leftharpoondown
7202     "2A = {100,100}, % \rightharpoonup

```



```

7203 "2B = {100,100}, % \rightharpoondown
7204 "2C = {300,200}, % \lhook
7205 "2D = {200,300}, % \rhook
7206 "2E = { ,100}, % \triangleright
7207 "2F = {100, }, % \triangleleft
7208 "3A = { ,500}, % ., \ldotp
7209 "3B = { ,500}, % ,
7210 "3C = {200,100}, % <
7211 "3D = {300,400}, % /
7212 "3E = {100,200}, % >
7213 "3F = {200,200}, % \star
7214 "5B = { ,100}, % \flat
7215 "5E = {200,200}, % \smile
7216 "5F = {200,200}, % \frown
7217 "7C = {100, }, % \jmath
7218 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

7219 }
7220

```

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}

```

```

7221 \SetProtrusion
7222 [ name = cmr-math-symbols ]
7223 { encoding = OMS,
7224   family = cmsy,
7225   series = {m,b},
7226   shape = n }
7227 {
7228   A = {150, 50}, % \mathcal
7229   C = { ,100},
7230   D = { , 50},
7231   F = { 50,150},
7232   I = { ,100},
7233   J = {100,150},
7234   K = { ,100},
7235   L = {100, },
7236   M = { 50, 50},
7237   N = { 50,100},
7238   P = { , 50},
7239   Q = { 50, },
7240   R = { , 50},
7241   T = { 50,150},
7242   V = { 50, 50},
7243   W = { , 50},
7244   X = {100,100},
7245   Y = {100, },
7246   Z = {100,150},
7247   "00 = {300,300}, % -
7248   "01 = { ,700}, % \cdot, \cdotp
7249   "02 = {150,250}, % \times
7250   "03 = {150,250}, % *, \ast
7251   "04 = {200,300}, % \div
7252   "05 = {150,250}, % \diamond
7253   "06 = {200,200}, % \pm
7254   "07 = {200,200}, % \mp
7255   "08 = {100,100}, % \oplus
7256   "09 = {100,100}, % \ominus
7257   "0A = {100,100}, % \otimes
7258   "0B = {100,100}, % \oslash
7259   "0C = {100,100}, % \odot
7260   "0D = {100,100}, % \bigcirc

```

```

7261 "0E = {100,100}, % \circ
7262 "0F = {100,100}, % \bullet
7263 "10 = {100,100}, % \asymp
7264 "11 = {100,100}, % \equiv
7265 "12 = {200,100}, % \subseteq
7266 "13 = {100,200}, % \supseteq
7267 "14 = {200,100}, % \leq
7268 "15 = {100,200}, % \geq
7269 "16 = {200,100}, % \preceq
7270 "17 = {100,200}, % \succeq
7271 "18 = {200,200}, % \sim
7272 "19 = {150,150}, % \approx
7273 "1A = {200,100}, % \subset
7274 "1B = {100,200}, % \supset
7275 "1C = {200,100}, % \ll
7276 "1D = {100,200}, % \gg
7277 "1E = {300,100}, % \prec
7278 "1F = {100,300}, % \succ
7279 "20 = {100,200}, % \leftarrow
7280 "21 = {200,100}, % \rightarrow
7281 "22 = {100,100}, % \uparrow
7282 "23 = {100,100}, % \downarrow
7283 "24 = {100,100}, % \leftrightarrow
7284 "25 = {100,100}, % \nearrow
7285 "26 = {100,100}, % \searrow
7286 "27 = {100,100}, % \simeq
7287 "28 = {100,100}, % \Leftarrow
7288 "29 = {100,100}, % \Rightarrow
7289 "2A = {100,100}, % \Uparrow
7290 "2B = {100,100}, % \Downarrow
7291 "2C = {100,100}, % \Leftrightarrow
7292 "2D = {100,100}, % \nrightarrow
7293 "2E = {100,100}, % \swarrow
7294 "2F = { ,100}, % \propto
7295 "30 = { ,400}, % \prime
7296 "31 = {100,100}, % \infty
7297 "32 = {150,100}, % \in
7298 "33 = {100,150}, % \ni
7299 "34 = {100,100}, % \triangle, \bigtriangleup
7300 "35 = {100,100}, % \bigtriangledown
7301 "38 = { ,100}, % \forall
7302 "39 = {100, }, % \exists
7303 "3A = {200, }, % \neg
7304 "3E = {200,200}, % \top
7305 "3F = {200,200}, % \bot, \perp
7306 "5E = {100,200}, % \wedge
7307 "5F = {100,200}, % \vee
7308 "60 = { ,300}, % \vdash
7309 "61 = {300, }, % \dashv
7310 "62 = {100,100}, % \lfloor
7311 "63 = {100,100}, % \rfloor
7312 "64 = {100,100}, % \lceil
7313 "65 = {100,100}, % \rceil
7314 "66 = {150, }, % \lbrace
7315 "67 = { ,150}, % \rbrace
7316 "68 = {400, }, % \langle
7317 "69 = { ,400}, % \rangle
7318 "6C = {100,100}, % \updownarrow
7319 "6D = {100,100}, % \Updownarrow
7320 "6E = {100,300}, % \, \backslash, \setminus
7321 "72 = {100,100}, % \nabla
7322 "79 = {200,200}, % \dagger
7323 "7A = {100,100}, % \ddagger
7324 "7B = {100, }, % \mathparagraph
7325 "7C = {100,100}, % \clubsuit

```

```

7326 "7D = {100,100}, % \diamondsuit
7327 "7E = {100,100}, % \heartsuit
7328 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

7329 }
7330

```

We don't bother about 'largesymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
```

```

7331 </cmr>
7332 </cfg-t>

```

### 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
7333 <*cfg-u>
```

Symbol font 'a'.

```

7334 <*msa>
7335 \SetProtrusion
7336 [ name = AMS-a ]
7337 { encoding = U,
7338   family = msa }
7339 {
7340   "05 = {150,250}, % \centerdot
7341   "06 = {100,100}, % \lozenge
7342   "07 = { 50, 50}, % \blacklozenge
7343   "08 = { 50, 50}, % \circlearrowright
7344   "09 = { 50, 50}, % \circlearrowleft
7345   "0A = {100,100}, % \rightleftharpoons
7346   "0B = {100,100}, % \leftrightharpoons
7347   "0D = {-50,200}, % \Vdash
7348   "0E = {-50,200}, % \Vvdash
7349   "0F = {-70,150}, % \vDash
7350   "10 = {100,150}, % \twoheadrightarrow
7351   "11 = {100,150}, % \twoheadleftarrow
7352   "12 = { 50,100}, % \leftleftarrows
7353   "13 = { 50, 80}, % \rightrightarrows
7354   "14 = {120,120}, % \upuparrows
7355   "15 = {120,120}, % \downdownarrows
7356   "16 = {200,200}, % \upharpoonright
7357   "17 = {200,200}, % \downharpoonright
7358   "18 = {200,200}, % \upharpoonleft
7359   "19 = {200,200}, % \downharpoonleft
7360   "1A = { 80,100}, % \rightarrowtail
7361   "1B = { 80,100}, % \leftarrowtail
7362   "1C = { 50, 50}, % \leftrightarrows
7363   "1D = { 50, 50}, % \rightleftarrows
7364   "1E = {250, }, % \Lsh
7365   "1F = { ,250}, % \Rsh
7366   "20 = {100,100}, % \rightsquigarrow
7367   "21 = {100,100}, % \leftrightsquigarrow
7368   "22 = {100, 50}, % \looparrowleft
7369   "23 = { 50,100}, % \looparrowright
7370   "24 = { 50, 80}, % \circeq
7371   "25 = { ,100}, % \succsim
7372   "26 = { ,100}, % \gtrsim
7373   "27 = { ,100}, % \gtrapprox
7374   "28 = {150, 50}, % \multimap
7375   "2B = {100,150}, % \doteqdot

```

```

7376 "2C = {100,150}, % \triangleq
7377 "2D = {100, 50}, % \precsim
7378 "2E = {100, 50}, % \lessim
7379 "2F = { 50, 50}, % \lessapprox
7380 "30 = {100, 50}, % \eqslantless
7381 "31 = { 50, 50}, % \eqslantgtr
7382 "32 = {100, 50}, % \curlyeqprec
7383 "33 = { 50,100}, % \curlyeqsucc
7384 "34 = {100, 50}, % \preccurlyeq
7385 "36 = { 50, }, % \leqslant
7386 "38 = { , 50}, % \backprime
7387 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
7388 "3C = { 50,100}, % \succcurlyeq
7389 "3E = { , 50}, % \geqslant
7390 "40 = { , 50}, % \sqsubset
7391 "41 = { 50, }, % \sqsupset
7392 "42 = { ,150}, % \vartriangleright, \rhd
7393 "43 = {150, }, % \vartriangleleft, \lhd
7394 "44 = { ,100}, % \trianglerighteq, \unrhd
7395 "45 = {100, }, % \trianglelefteq, \unlhd
7396 "46 = {100,100}, % \bigstar
7397 "48 = { 50, 50}, % \blacktriangledown
7398 "49 = { ,100}, % \blacktriangleright
7399 "4A = {100, }, % \blacktriangleleft
7400 "4B = { ,150}, % \dashrightarrow (the arrow)
7401 "4C = {150, }, % \dashleftarrow
7402 "4D = { 50, 50}, % \vartriangle
7403 "4E = { 50, 50}, % \blacktriangle
7404 "4F = { 50, 50}, % \triangledown
7405 "50 = { 50, 50}, % \eqcirc
7406 "56 = { ,150}, % \rightarrow
7407 "57 = {150, }, % \leftarrow
7408 "58 = {100,300}, % \checkmark
7409 "5C = { 50, 50}, % \angle
7410 "5D = { 50, 50}, % \measuredangle
7411 "5E = { 50, 50}, % \sphericalangle
7412 "5F = { , 50}, % \varpropto
7413 "60 = {100,100}, % \smallsmile
7414 "61 = {100,100}, % \smallfrown
7415 "62 = { 50, }, % \Subset
7416 "63 = { , 50}, % \Supset
7417 "66 = {150,150}, % \curlywedge
7418 "67 = {150,150}, % \curlyvee
7419 "68 = { 50,150}, % \leftthreetimes
7420 "69 = {100, 50}, % \rightthreetimes
7421 "6C = { 50, 50}, % \bumpeq
7422 "6D = { 50, 50}, % \Bumpeq
7423 "6E = {100, }, % \lll
7424 "6F = { ,100}, % \ggg
7425 "70 = { 50,100}, % \ulcorner
7426 "71 = {100, 50}, % \urcorner
7427 "75 = {150,200}, % \dotplus
7428 "76 = { 50,100}, % \backsim
7429 "78 = { 50,100}, % \llcorner
7430 "79 = {100, 50}, % \lrcorner
7431 "7C = {100,100}, % \intercal
7432 "7D = { 50, 50}, % \circledcirc
7433 "7E = { 50, 50}, % \circledast
7434 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

7435 }
7436
7437 (/msa)

```

Symbol font 'b'.

```

7438 (*msb)
7439 \SetProtrusion
7440 [ name      = AMS-b ]
7441 { encoding = U,
7442   family   = msb }
7443 {
7444   A = { 50, 50}, % \mathbb
7445   C = { 50, 50},
7446   G = {   , 50},
7447   L = {   , 50},
7448   P = {   , 50},
7449   R = {   , 50},
7450   T = {   , 50},
7451   V = { 50, 50},
7452   X = { 50, 50},
7453   Y = { 50, 50},
7454   "00 = { 50, 50}, % \lvertneqq
7455   "01 = { 50, 50}, % \gvertneqq
7456   "02 = { 50, 50}, % \nleq
7457   "03 = { 50, 50}, % \ngeq
7458   "04 = {100, 50}, % \nless
7459   "05 = { 50,150}, % \ngtr
7460   "06 = {100, 50}, % \nprec
7461   "07 = { 50,150}, % \nsucc
7462   "08 = { 50, 50}, % \lneqq
7463   "09 = { 50, 50}, % \gneqq
7464   "0A = {100,100}, % \leqslant
7465   "0B = {100,100}, % \geqslant
7466   "0C = {100, 50}, % \lneq
7467   "0D = { 50,100}, % \gneq
7468   "0E = {100, 50}, % \npreceq
7469   "0F = { 50,100}, % \nsucceq
7470   "10 = { 50,   }, % \precnsim
7471   "11 = { 50, 50}, % \succnsim
7472   "12 = { 50, 50}, % \lnsim
7473   "13 = { 50, 50}, % \gnsim
7474   "14 = { 50, 50}, % \lneqq
7475   "15 = { 50, 50}, % \ngeqq
7476   "16 = { 50, 50}, % \precneqq
7477   "17 = { 50, 50}, % \succneqq
7478   "18 = { 50, 50}, % \precnapprox
7479   "19 = { 50, 50}, % \succnapprox
7480   "1A = { 50, 50}, % \lnapprox
7481   "1B = { 50, 50}, % \gnapprox
7482   "1C = {150,200}, % \nsim
7483   "1D = { 50, 50}, % \ncong
7484   "1E = {100,150}, % \diagup
7485   "1F = {100,150}, % \diagdown
7486   "20 = {100, 50}, % \varsubsetneq
7487   "21 = { 50,100}, % \varsupsetneq
7488   "22 = {100, 50}, % \subsetneqq
7489   "23 = { 50,100}, % \supsetneqq
7490   "24 = {100, 50}, % \subsetneqq
7491   "25 = { 50,100}, % \supsetneqq
7492   "26 = {100, 50}, % \varsubsetneqq
7493   "27 = { 50,100}, % \varsupsetneqq
7494   "28 = {100, 50}, % \subsetneq
7495   "29 = { 50,100}, % \supsetneq
7496   "2A = {100, 50}, % \subseteq
7497   "2B = { 50,100}, % \supseteq
7498   "2C = { 50,100}, % \parallel
7499   "2D = {100,150}, % \mid
7500   "2E = {150,150}, % \shortmid
7501   "2F = {100,100}, % \shortparallel

```

```

7502 "30 = { ,150}, % \nvDash
7503 "31 = { ,150}, % \nVDash
7504 "32 = { ,100}, % \nvDash
7505 "33 = { ,100}, % \nVDash
7506 "34 = { ,100}, % \ntrianglerighteq
7507 "35 = {100, }, % \ntrianglelefteq
7508 "36 = {100, }, % \ntriangleleft
7509 "37 = { ,100}, % \ntriangleright
7510 "38 = {100,200}, % \leftarrow
7511 "39 = {100,200}, % \rightarrow
7512 "3A = {100,100}, % \Leftarrow
7513 "3B = { 50,100}, % \Rightarrow
7514 "3C = {100,100}, % \Leftrightarrow
7515 "3D = {100,200}, % \leftrightarrows
7516 "3E = { 50, 50}, % \divideontimes
7517 "3F = { 50, 50}, % \varnothing
7518 "60 = {200, }, % \Finv
7519 "61 = { , 50}, % \Game
7520 "68 = {100,100}, % \eqsim
7521 "69 = { 50, }, % \beth
7522 "6A = { 50, }, % \gimel
7523 "6B = {150, }, % \daleth
7524 "6C = {200, }, % \lessdot
7525 "6D = { ,200}, % \gtrdot
7526 "6E = {100,200}, % \ltimes
7527 "6F = {150,100}, % \rtimes
7528 "70 = { 50,100}, % \shortmid
7529 "71 = { 50, 50}, % \shortparallel
7530 "72 = {200,300}, % \smallsetminus
7531 "73 = {100,200}, % \thicksim
7532 "74 = { 50,100}, % \thickapprox
7533 "75 = { 50, 50}, % \approx
7534 "76 = { 50,100}, % \succapprox
7535 "77 = { 50, 50}, % \precapprox
7536 "78 = {100,100}, % \curvearrowleft
7537 "79 = { 50,150}, % \curvearrowright
7538 "7A = { 50,200}, % \digamma
7539 "7B = {100, 50}, % \varkappa
7540 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

7541 }
7542
7543 </msb>

```

### 15.8.8 Euler

Euler Roman font (package euler).

```

7544 <*eur>
7545 \SetProtrusion
7546 [ name = euler ]
7547 { encoding = U,
7548   family = eur }
7549 {
7550 "01 = {100,100},
7551 "03 = {100,150},
7552 "06 = { ,100},
7553 "07 = {100,150},
7554 "08 = {100,100},
7555 "0A = {100,100},
7556 "0B = { , 50},
7557 "0C = { ,100},
7558 "0D = {100,100},
7559 "0E = { ,100},

```

```

7560 "0F = {100,100},
7561 "10 = {100,100},
7562 "13 = { ,100},
7563 "14 = { ,100},
7564 "15 = { , 50},
7565 "16 = { , 50},
7566 "17 = { 50,100},
7567 "18 = { 50,100},
7568 "1A = { , 50},
7569 "1B = { , 50},
7570 "1C = { 50,100},
7571 "1D = { 50,100},
7572 "1E = { 50,100},
7573 "1F = { 50,100},
7574 "20 = { , 50},
7575 "21 = { , 50},
7576 "22 = { 50,100},
7577 "24 = { , 50},
7578 "27 = { 50,100},
7579 1 = {100,100},
7580 7 = { 50,100},
7581 "3A = {300,500},
7582 "3B = {200,400},
7583 "3C = {200,100},
7584 "3D = {200,200},
7585 "3E = {100,200},
7586 A = { ,100},
7587 D = { , 50},
7588 J = { 50, },
7589 K = { , 50},
7590 L = { , 50},
7591 Q = { , 50},
7592 T = { 50, },
7593 X = { 50, 50},
7594 Y = { 50, },
7595 h = { , 50},
7596 k = { , 50}
7597 }
7598

```

Extended by the `eulervm` package.

```

7599 \SetProtrusion
7600 [ name = euler-vm,
7601 load = euler ]
7602 { encoding = U,
7603 family = zeur }
7604 {
7605 "28 = {100,200},
7606 "29 = {100,200},
7607 "2A = {100,150},
7608 "2B = {100,150},
7609 "2C = {200,300},
7610 "2D = {200,300},
7611 "2E = { ,100},
7612 "2F = {100, },
7613 "3F = {150,150},
7614 "5B = { ,100},
7615 "5E = {100,100},
7616 "5F = {100,100},
7617 "80 = { , 50},
7618 "81 = {200,250},
7619 "82 = {100,200}
7620 }
7621
7622 </eur>

```

## Euler Script font (euca1).

```

7623 (*eus)
7624 \SetProtrusion
7625 [ name      = euscript ]
7626 { encoding = U,
7627   family   = eus   }
7628 {
7629     A = {100,100},
7630     B = { 50,100},
7631     C = { 50, 50},
7632     D = { 50,100},
7633     E = { 50,100},
7634     F = { 50,  },
7635     G = { 50,  },
7636     H = {  ,100},
7637     K = {  , 50},
7638     L = {  ,150},
7639     M = {  , 50},
7640     N = {  , 50},
7641     O = { 50, 50},
7642     P = { 50, 50},
7643     T = {  ,100},
7644     U = {  , 50},
7645     V = { 50, 50},
7646     W = { 50, 50},
7647     X = { 50, 50},
7648     Y = { 50,  },
7649     Z = { 50,100},
7650     "00 = {250,250},
7651     "18 = {200,200},
7652     "3A = {200,150},
7653     "40 = {  ,100},
7654     "5E = {100,100},
7655     "5F = {100,100},
7656     "66 = { 50,  },
7657     "67 = {  , 50},
7658     "6E = {200,200}
7659 }
7660
7661 \SetProtrusion
7662 [ name      = euscript-vm,
7663   load      = euscript ]
7664 { encoding = U,
7665   family   = zeus   }
7666 {
7667     "01 = {600,600},
7668     "02 = {200,200},
7669     "03 = {200,200},
7670     "04 = {200,200},
7671     "05 = {150,150},
7672     "06 = {200,200},
7673     "07 = {200,200},
7674     "08 = {100,100},
7675     "09 = {100,100},
7676     "0A = {100,100},
7677     "0B = {100,100},
7678     "0C = {100,100},
7679     "0D = {100,100},
7680     "0E = {150,150},
7681     "0F = {100,100},
7682     "10 = {150,150},
7683     "11 = {100,100},
7684     "12 = {150,100},
7685     "13 = {100,150},
7686     "14 = {150,100},

```



```
7687 "15 = {100,150},
7688 "16 = {200,100},
7689 "17 = {100,200},
7690 "19 = {150,150},
7691 "1A = {150,100},
7692 "1B = {100,150},
7693 "1C = {100,100},
7694 "1D = {100,100},
7695 "1E = {250,100},
7696 "1F = {100,250},
7697 "20 = {150,200},
7698 "21 = {150,200},
7699 "22 = {150,150},
7700 "23 = {150,150},
7701 "24 = {100,200},
7702 "25 = {150,150},
7703 "26 = {150,150},
7704 "27 = {100,100},
7705 "28 = {100,100},
7706 "29 = {100,150},
7707 "2A = {100,100},
7708 "2B = {100,100},
7709 "2C = {100,100},
7710 "2D = {150,150},
7711 "2E = {150,150},
7712 "2F = {100,100},
7713 "30 = {100,100},
7714 "31 = {100,100},
7715 "32 = {100,100},
7716 "33 = {100,100},
7717 "34 = {100,100},
7718 "35 = {100,100},
7719 "3E = {150,150},
7720 "3F = {150,150},
7721 "60 = { ,200},
7722 "61 = {200, },
7723 "62 = {100,100},
7724 "63 = {100,100},
7725 "64 = {100,100},
7726 "65 = {100,100},
7727 "68 = {300, },
7728 "69 = { ,300},
7729 "6C = {100,100},
7730 "6D = {100,100},
7731 "6F = {100,100},
7732 "72 = {100,100},
7733 "73 = {200,100},
7734 "76 = { ,100},
7735 "77 = {100, },
7736 "78 = { 50, 50},
7737 "79 = {100,100},
7738 "7A = {100,100},
7739 "7D = {150,150},
7740 "7E = {100,100},
7741 "A8 = {100,100},
7742 "A9 = {100,100},
7743 "AB = {200,200},
7744 "BA = { ,200},
7745 "BB = { ,200},
7746 "BD = {200,200},
7747 "DE = {200,200}
7748 }
7749
7750 (/eus)
```

## Euler Fraktur font (eufrak).

```

7751 <*euf>
7752 \SetProtrusion
7753   [ name      = mathfrak ]
7754   { encoding = U,
7755     family   = euf  }
7756   {
7757     A = { , 50},
7758     B = { , 50},
7759     C = { 50, 50},
7760     D = { , 80},
7761     E = { 50, },
7762     G = { , 50},
7763     L = { , 80},
7764     O = { , 50},
7765     T = { , 80},
7766     X = { 80, 50},
7767     Z = { 80, 50},
7768     b = { , 50},
7769     c = { , 50},
7770     k = { , 50},
7771     p = { , 50},
7772     q = { 50, },
7773     v = { , 50},
7774     w = { , 50},
7775     x = { , 50},
7776     1 = {100,100},
7777     2 = { 80, 80},
7778     3 = { 80, 50},
7779     4 = { 80, 50},
7780     7 = { 50, 50},
7781     "12 = {500,500},
7782     "13 = {500,500},
7783     ! = { ,200},
7784     ' = {200,300},
7785     ( = {200, },
7786     ) = { ,200},
7787     * = {200,200},
7788     + = {200,250},
7789     - = {200,200},
7790     {,} = {300,300},
7791     . = {400,400},
7792     {=} = {200,200},
7793     : = { ,200},
7794     ; = { ,200},
7795     ] = { ,200}
7796   }
7797
7798 </euf>
7799 </cfg-u>

```

**15.8.9 Euro symbols**

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroi tc) and marvosym<sup>24</sup>). The euroi tc settings are hidden in the package itself (14.3.7) for ‘free software’ compliance reasons. (Not quite sure whether this is what Karl really had in mind ...)

```

7800 <*cfg-e>
7801 \SetProtrusion
7802 <zpeu> { encoding = U,
7803 <mvs> { encoding = {OT1,U},

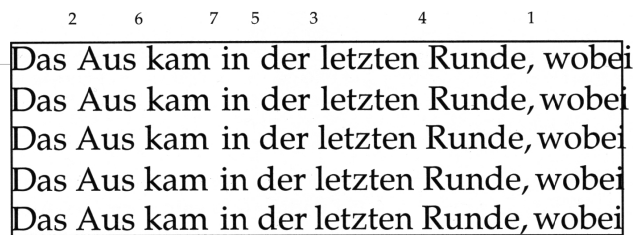
```

---

24 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.



```

7804 <zpeu>    family = zpeu }
7805 <mvs>     family = mvs }
7806 {
7807 <zpeu>     E = {50, }
7808 <mvs>     164 = {50,50}, % \EUR
7809 <mvs>     068 = {50,-100} % \EURdig
7810 }
7811
7812 <*zpeu>
7813 \SetProtrusion
7814 { encoding = U,
7815   family = zpeu,
7816   shape = it* }
7817 {
7818   E = {100,-50}
7819 }
7820
7821 \SetProtrusion
7822 { encoding = U,
7823   family = {zpeus,eurosans} }
7824 {
7825   E = {100,50}
7826 }
7827
7828 \SetProtrusion
7829 { encoding = U,
7830   family = {zpeus,eurosans},
7831   shape = it* }
7832 {
7833   E = {200, }
7834 }
7835
7836 </zpeu>
7837 </cfg-e>

```

## 15.9 Interword spacing

Default unit is space.

```

7838 <*m-t|cmr>
7839 %%% -----
7840 %%% INTERWORD SPACING
7841
7842 </m-t|cmr>
7843 <*m-t>
7844 \SetExtraSpacing
7845 [ name = default ]
7846 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7847 {

```

These settings are only a first approximation. The following reasoning is from a 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

7848           { , } = { , -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)

7849           r = { , -300, 300 } ,

- [before or] after lowercase characters with ascenders

7850           b = { , -200, 200 } ,

7851           d = { , -200, 200 } ,

7852           f = { , -200, 200 } ,

7853           h = { , -200, 200 } ,

7854           k = { , -200, 200 } ,

7855           l = { , -200, 200 } ,

7856           t = { , -200, 200 } ,

- [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., ‘v’, or ‘w’

7857           c = { , -100, 100 } ,

7858           p = { , -100, 100 } ,

7859           v = { , -100, 100 } ,

7860           w = { , -100, 100 } ,

7861           z = { , -100, 100 } ,

7862           x = { , -100, 100 } ,

7863           y = { , -100, 100 } ,

- [before or] after lowercase characters with x-height plus descender without additional optical space

7864           i = { , 50, -50 } ,

7865           m = { , 50, -50 } ,

7866           n = { , 50, -50 } ,

7867           u = { , 50, -50 } ,

- after colon and semicolon

7868           : = { , 200, -200 } ,

7869           ; = { , 200, -200 } ,

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

7870           . = { , 250, -250 } ,

7871           ! = { , 250, -250 } ,

7872           ? = { , 250, -250 } ,

The order has to be reversed when enlarging is needed.’

7873           }

7874

7875  $\langle /m-t \rangle$

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.<sup>25</sup>

```
7876 (*cmr*)
7877 \SetExtraSpacing
7878   [ name      = T2A,
7879     load      = default ]
7880   { encoding = T2A,
7881     family   = cmr }
7882   {
7883     \cyrg = { , -300, 300 },
7884     \cyrb = { , -200, 200 },
7885     \cyrk = { , -200, 200 },
7886     \cyrs = { , -100, 100 },
7887     \cyrr = { , -100, 100 },
7888     \cyrh = { , -100, 100 },
7889     \cyru = { , -100, 100 },
7890     \cyrt = { , 50, -50 },
7891     \cyrp = { , 50, -50 },
7892     \cyri = { , 50, -50 },
7893     \cyrishrt = { , 50, -50 },
7894   }
7895
```

### 15.9.1 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 T<sub>E</sub>Xbook:

‘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’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i.e., 333.

```
7896 \SetExtraSpacing
7897   [ name      = nonfrench-cmr,
7898     load      = default,
7899     context   = nonfrench ]
7900   { encoding = {OT1,T1,LY1,OT4,QX,T5},
```

---

25 Contributed by *Karl Karlsson*.

```

7901     family    = cmr }
7902   {

```

latex.ltx has:

```

\def\nonfrenchspacing{
  \sfcode`. 3000
  \sfcode`? 3000
  \sfcode`! 3000

```

```

7903   . = {333,2000,-667},
7904   ? = {333,2000,-667},
7905   ! = {333,2000,-667},

```

```

\sfcodes\ : 2000

```

```

7906   : = {333,1000,-500},

```

```

\sfcodes\ ; 1500

```

```

7907   ; = {   , 500,-333},

```

```

\sfcodes\ , 1250

```

```

7908   {,}= {   , 250,-200}

```

```

}

```

```

7909   }
7910
7911 </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.

```

7912 < *m-t >
7913 \SetExtraSpacing
7914 [ name      = nonfrench-default,
7915   load      = default,
7916   context   = nonfrench ]
7917 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7918 {
7919   . = {240,2000,-667},
7920   ? = {240,2000,-667},
7921   ! = {240,2000,-667},
7922   : = {240,1000,-500},
7923   ; = {   , 500,-333},
7924   {,}= {   , 250,-200}
7925 }
7926

```

Empty settings to prevent spurious warnings.

```

7927 \SetExtraSpacing
7928 [ name = empty ]
7929 { encoding = {TS1} }
7930 { }
7931

```

## 15.10 Additional kerning

Default unit is 1 em.

```

7932 %%% -----
7933 %%% ADDITIONAL KERNING
7934

```

A dummy list to be loaded when no context is active.

```

7935 \SetExtraKerning
7936   [ name = empty ]
7937   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
7938   { }
7939

```

### 15.10.1 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. (Wikipedia<sup>26</sup> 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.)

```

7940 \SetExtraKerning
7941   [ name      = french-default,
7942     context   = french,
7943     unit      = space ]
7944   { encoding = {OT1,T1,LY1} }
7945   {
7946     : = {1000,}, % = \fontdimen2
7947     ; = {500, }, % ~ \thinspace
7948     ! = {500, },
7949     ? = {500, }
7950   }
7951

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTeX.

```

7952 \SetExtraKerning
7953   [ name      = french-guillemets,
7954     context   = french-guillemets,
7955     load      = french-default,
7956     unit      = space ]
7957   { encoding = {T1,LY1} }
7958   {
7959     \guillemotleft = { ,800}, % = 0.8\fontdimen2
7960     \guillemotright = {800, }
7961   }
7962
7963 \SetExtraKerning
7964   [ name      = french-guillemets-OT1,
7965     context   = french-guillemets,
7966     load      = french-default,
7967     unit      = space ]
7968   { encoding = OT1 }
7969   { }
7970

```

### 15.10.2 Turkish

```

7971 \SetExtraKerning
7972   [ name      = turkish,
7973     context   = turkish ]
7974   { encoding = {OT1,T1,LY1} }

```

---

26 [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique), 5 July 2007.

```
7975     {  
7976         : = {167, }, % = \thinspace  
7977         ! = {167, },  
7978         {=} = {167, }  
7979     }  
7980  
7981 </m-t>  
7982 </config>
```






[illegible]

8087 S}, % Cyr  
 8088 T = {T, T̃, T̂, T̄, T̅, T̆, Ṫ, T̈, T̉, T̊, T̋, Ť, T̍, T̎, T̏, T̐, T̑, T̒, T̓, T̔, T̕, T̖, T̗, T̘, T̙, T̚, T̛, T̜, T̝, T̞, T̟, T̠, T̡, T̢, Ṭ, T̤, T̥, Ț, Ţ, T̨, T̩, T̪, T̫, T̬, Ṱ, T̮, T̯, T̰, Ṯ, T̲, T̳, T̴, T̵, T̶, T̷, T̸, T̹, T̺, T̻, T̼, T̽, T̾, T̿, T̿̂, T̿̃, T̿̄, T̿̅, T̿̆, T̿̇, T̿̈, T̿̉, T̿̊, T̿̋, T̿̌, T̿̍, T̿̎, T̿̏, T̿̐, T̿̑, T̿̒, T̿̓, T̿̔, T̿̕, T̖̿, T̗̿, T̘̿, T̙̿, T̿̚, T̛̿, T̜̿, T̝̿, T̞̿, T̟̿, T̠̿, T̡̿, T̢̿, Ṭ̿, T̤̿, T̥̿, Ț̿, Ţ̿, T̨̿, T̩̿, T̪̿, T̫̿, T̬̿, Ṱ̿, T̮̿, T̯̿, T̰̿, Ṯ̿, T̲̿, T̳̿, T̴̿, T̵̿, T̶̿, T̷̿, T̸̿, T̹̿, T̺̿, T̻̿, T̼̿, T̿̽, T̿̾, T̿̿},  
 8089 T, T̃}, % Cyr  
 8090 U = {U, Ů, Ű, Ū, Ŭ, Ů̃, Ů̂, Ů̄, Ů̅, Ů̆, Ů̇, Ů̈, Ů̉, Ů̊, Ů̋, Ů̌, Ů̍, Ů̎, Ů̏, Ů̐, Ů̑, Ů̒, Ů̓, Ů̔, Ů̕, Ů̖, Ů̗, Ů̘, Ů̙, Ů̚, Ư̊, Ů̜, Ů̝, Ů̞, Ů̟, Ů̠, Ů̡, Ů̢, Ụ̊, Ṳ̊, Ů̥, Ů̦, Ů̧, Ų̊, Ů̩, Ů̪, Ů̫, Ů̬, Ṷ̊, Ů̮, Ů̯, Ṵ̊, Ů̱, Ů̲, Ů̳, Ů̴, Ů̵, Ů̶, Ů̷, Ů̸, Ů̹, Ů̺, Ů̻, Ů̼, Ů̽, Ů̾, Ů̿},  
 8091 V = {Ṽ, V̂},  
 8092 W = {Ŵ, ŵ, W̃, Ŵ, W̄, W̅, W̆, Ẇ, Ẅ, W̉, W̊, W̋, W̌, W̍, W̎, W̏, W̐, W̑, W̒, W̓, W̔, W̕, W̖, W̗, W̘, W̙, W̚, W̛, W̜, W̝, W̞, W̟, W̠, W̡, W̢, Ẉ, W̤, W̥, W̦, W̧, W̨, W̩, W̪, W̫, W̬, W̭, W̮, W̯, W̰, W̱, W̲, W̳, W̴, W̵, W̶, W̷, W̸, W̹, W̺, W̻, W̼, W̽, W̾, W̿},  
 8093 W}, % Cyr  
 8094 X = {X̃, X̂},  
 8095 X, X̃, X̂, X̄}, % Cyr  
 8096 Y = {Ỹ, Ŷ, Ȳ, Y̅, Y̆, Ẏ, Ÿ, Ỷ, Y̊, Y̋, Y̌, Y̍, Y̎, Y̏, Y̐, Y̑, Y̒, Y̓, Y̔, Y̕, Y̖, Y̗, Y̘, Y̙, Y̚, Y̛, Y̜, Y̝, Y̞, Y̟, Y̠, Y̡, Y̢, Ỵ, Y̤, Y̥, Y̦, Y̧, Y̨, Y̩, Y̪, Y̫, Y̬, Y̭, Y̮, Y̯, Y̰, Y̱, Y̲, Y̳, Y̴, Y̵, Y̶, Y̷, Y̸, Y̹, Y̺, Y̻, Y̼, Y̽, Y̾, Y̿},  
 8097 Y, Ỹ}, % Cyr  
 8098 Z = {Z̃, Ẑ, Z̄, Z̅, Z̆, Ż, Z̈, Z̉, Z̊, Z̋, Ž, Z̍, Z̎, Z̏, Z̐, Z̑, Z̒, Z̓, Z̔, Z̕, Z̖, Z̗, Z̘, Z̙, Z̚, Z̛, Z̜, Z̝, Z̞, Z̟, Z̠, Z̡, Z̢, Ẓ, Z̤, Z̥, Z̦, Z̧, Z̨, Z̩, Z̪, Z̫, Z̬, Z̭, Z̮, Z̯, Z̰, Ẕ, Z̲, Z̳, Z̴, Z̵, Z̶, Z̷, Z̸, Z̹, Z̺, Z̻, Z̼, Z̽, Z̾, Z̿},  
 8099 a = {ā, á, â, ã, ä, å, ã̃, ã̂, ã̄, ã̅, ã̆, ã̇, ã̈, ã̉, ã̊, ã̋, ã̌, ã̍, ã̎, ã̏, ã̐, ã̑, ã̒, ã̓, ã̔, ã̕, ã̖, ã̗, ã̘, ã̙, ã̚, ã̛, ã̜, ã̝, ã̞, ã̟, ã̠, ã̡, ã̢, ạ̃, ã̤, ḁ̃, ã̦, ã̧, ą̃, ã̩, ã̪, ã̫, ã̬, ã̭, ã̮, ã̯, ã̰, ã̱, ã̲, ã̳, ã̴, ã̵, ã̶, ã̷, ã̸, ã̹, ã̺, ã̻, ã̼, ã̽, ã̾, ã̿}, % a'  
 8100 a, ã̃}, % Cyr  
 8101 æ = {æ̃, æ̂},  
 8102 æ}, % Cyr  
 8103 b = {b̃, b̂, b̄},  
 8104 c = {ç̃, ç̂, ç̄, ç̅, ç̆, ç̇, ç̈, ç̉, ç̊, ç̋, ç̌, ç̍, ç̎, ç̏, ç̐, ç̑, ç̒, ç̓, ç̔, ç̕, ç̖, ç̗, ç̘, ç̙, ç̚, ç̛, ç̜, ç̝, ç̞, ç̟, ç̠, ç̡, ç̢, ç̣, ç̤, ç̥, ç̦, ç̧, ç̨, ç̩, ç̪, ç̫, ç̬, ç̭, ç̮, ç̯, ç̰, ç̱, ç̲, ç̳, ç̴, ç̵, ç̶, ç̷, ç̸, ç̹, ç̺, ç̻, ç̼, ç̽, ç̾, ç̿},  
 8105 c, ç̃}, % Cyr  
 8106 d = {đ̃, đ̂, đ̄, đ̅, đ̆, đ̇, đ̈, đ̉, đ̊, đ̋, đ̌, đ̍, đ̎, đ̏, đ̐, đ̑, đ̒, đ̓, đ̔, đ̕, đ̖, đ̗, đ̘, đ̙, đ̚, đ̛, đ̜, đ̝, đ̞, đ̟, đ̠, đ̡, đ̢, đ̣, đ̤, đ̥, đ̦, đ̧, đ̨, đ̩, đ̪, đ̫, đ̬, đ̭, đ̮, đ̯, đ̰, đ̱, đ̲, đ̳, đ̴, đ̵, đ̶, đ̷, đ̸, đ̹, đ̺, đ̻, đ̼, đ̽, đ̾, đ̿},  
 8107 e = {è̃, è̂, è̄, è̅, è̆, è̇, è̈, è̉, è̊, è̋, è̌, è̍, è̎, è̏, è̐, è̑, è̒, è̓, è̔, è̕, è̖, è̗, è̘, è̙, è̚, è̛, è̜, è̝, è̞, è̟, è̠, è̡, è̢, ẹ̀, è̤, è̥, è̦, ȩ̀, ę̀, è̩, è̪, è̫, è̬, ḙ̀, è̮, è̯, ḛ̀, è̱, è̲, è̳, è̴, è̵, è̶, è̷, è̸, è̹, è̺, è̻, è̼, è̽, è̾, è̿},  
 8108 e, è̃}, % Cyr  
 8109 f = {f̃, f̂, f̄}, % /f f  
 8110 g = {ġ̃, ġ̂, ġ̄, ġ̅, ġ̆, ġ̇, ġ̈, ġ̉, ġ̊, ġ̋, ġ̌, ġ̍, ġ̎, ġ̏, ġ̐, ġ̑, ġ̒, ġ̓, ġ̔, ġ̕, ġ̖, ġ̗, ġ̘, ġ̙, ġ̚, ġ̛, ġ̜, ġ̝, ġ̞, ġ̟, ġ̠, ġ̡, ġ̢, ġ̣, ġ̤, ġ̥, ġ̦, ģ̇, ġ̨, ġ̩, ġ̪, ġ̫, ġ̬, ġ̭, ġ̮, ġ̯, ġ̰, ġ̱, ġ̲, ġ̳, ġ̴, ġ̵, ġ̶, ġ̷, ġ̸, ġ̹, ġ̺, ġ̻, ġ̼, ġ̽, ġ̾, ġ̿},  
 8111 h = {h̃, ĥ, h̄, h̅, h̆, ḣ, ḧ, h̉, h̊, h̋, ȟ, h̍, h̎, h̏, h̐, h̑, h̒, h̓, h̔, h̕, h̖, h̗, h̘, h̙, h̚, h̛, h̜, h̝, h̞, h̟, h̠, h̡, h̢, ḥ, h̤, h̥, h̦, ḩ, h̨, h̩, h̪, h̫, h̬, h̭, ḫ, h̯, h̰, ẖ, h̲, h̳, h̴, h̵, h̶, h̷, h̸, h̹, h̺, h̻, h̼, h̽, h̾, h̿},  
 8112 h, h̃}, % Cyr  
 8113 i = {ı̃, ı̂, ı̄, ı̅, ı̆, ı̇, ı̈, ı̉, ı̊, ı̋, ı̌, ı̍, ı̎, ı̏, ı̐, ı̑, ı̒, ı̓, ı̔, ı̕, ı̖, ı̗, ı̘, ı̙, ı̚, ı̛, ı̜, ı̝, ı̞, ı̟, ı̠, ı̡, ı̢, ı̣, ı̤, ı̥, ı̦, ı̧, ı̨, ı̩, ı̪, ı̫, ı̬, ı̭, ı̮, ı̯, ı̰, ı̱, ı̲, ı̳, ı̴, ı̵, ı̶, ı̷, ı̸, ı̹, ı̺, ı̻, ı̼, ı̽, ı̾, ı̿},  
 8114 i, ı̃}, % Cyr  
 8115 j = {j̃, ĵ, j̄},  
 8116 j}, % Cyr  
 8117 k = {k̃, k̂, k̄, k̅, k̆, k̇, k̈, k̉, k̊, k̋, ǩ, k̍, k̎, k̏, k̐, k̑, k̒, k̓, k̔, k̕, k̖, k̗, k̘, k̙, k̚, k̛, k̜, k̝, k̞, k̟, k̠, k̡, k̢, ḳ, k̤, k̥, k̦, ķ, k̨, k̩, k̪, k̫, k̬, k̭, k̮, k̯, k̰, ḵ, k̲, k̳, k̴, k̵, k̶, k̷, k̸, k̹, k̺, k̻, k̼, k̽, k̾, k̿},  
 8118 l = {ĺ̃, ĺ̂, ĺ̄, ĺ̅, ĺ̆, ĺ̇, ĺ̈, ĺ̉, ĺ̊, ĺ̋, ĺ̌, ĺ̍, ĺ̎, ĺ̏, ĺ̐, ĺ̑, ĺ̒, ĺ̓, ĺ̔, ĺ̕, ĺ̖, ĺ̗, ĺ̘, ĺ̙, ĺ̚, ĺ̛, ĺ̜, ĺ̝, ĺ̞, ĺ̟, ĺ̠, ĺ̡, ĺ̢, ḷ́, ĺ̤, ĺ̥, ĺ̦, ļ́, ĺ̨, ĺ̩, ĺ̪, ĺ̫, ĺ̬, ḽ́, ĺ̮, ĺ̯, ĺ̰, ḻ́, ĺ̲, ĺ̳, ĺ̴, ĺ̵, ĺ̶, ĺ̷, ĺ̸, ĺ̹, ĺ̺, ĺ̻, ĺ̼, ĺ̽, ĺ̾, ĺ̿},  
 8119 m = {m̃, m̂, m̄},  
 8120 n = {ñ̃, ñ̂, ñ̄, ñ̅, ñ̆, ñ̇, ñ̈, ñ̉, ñ̊, ñ̋, ñ̌, ñ̍, ñ̎, ñ̏, ñ̐, ñ̑, ñ̒, ñ̓, ñ̔, ñ̕, ñ̖, ñ̗, ñ̘, ñ̙, ñ̚, ñ̛, ñ̜, ñ̝, ñ̞, ñ̟, ñ̠, ñ̡, ñ̢, ṇ̃, ñ̤, ñ̥, ñ̦, ņ̃, ñ̨, ñ̩, ñ̪, ñ̫, ñ̬, ṋ̃, ñ̮, ñ̯, ñ̰, ṉ̃, ñ̲, ñ̳, ñ̴, ñ̵, ñ̶, ñ̷, ñ̸, ñ̹, ñ̺, ñ̻, ñ̼, ñ̽, ñ̾, ñ̿},  
 8121 o = {ỗ, ô̂, ô̄, ô̅, ô̆, ô̇, ô̈, ổ, ô̊, ô̋, ô̌, ô̍, ô̎, ô̏, ô̐, ô̑, ô̒, ô̓, ô̔, ô̕, ô̖, ô̗, ô̘, ô̙, ô̚, ơ̂, ô̜, ô̝, ô̞, ô̟, ô̠, ô̡, ô̢, ộ, ô̤, ô̥, ô̦, ô̧, ǫ̂, ô̩, ô̪, ô̫, ô̬, ô̭, ô̮, ô̯, ô̰, ô̱, ô̲, ô̳, ô̴, ô̵, ô̶, ô̷, ô̸, ô̹, ô̺, ô̻, ô̼, ô̽, ô̾, ô̿},  
 8122 o, ỗ}, % Cyr  
 8123 p = {p̃, p̂, p̄},  
 8124 p, p̃}, % Cyr  
 8125 q = {q̃, q̂, q̄}, % Cyr  
 8126 r = {r̃, r̂, r̄, r̅, r̆, ṙ, r̈, r̉, r̊, r̋, ř, r̍, r̎, ȑ, r̐, ȓ, r̒, r̓, r̔, r̕, r̖, r̗, r̘, r̙, r̚, r̛, r̜, r̝, r̞, r̟, r̠, r̡, r̢, ṛ, r̤, r̥, r̦, ŗ, r̨, r̩, r̪, r̫, r̬, r̭, r̮, r̯, r̰, ṟ, r̲, r̳, r̴, r̵, r̶, r̷, r̸, r̹, r̺, r̻, r̼, r̽, r̾, r̿},  
 8127 s = {s̃, ŝ, s̄, s̅, s̆, ṡ, s̈, s̉, s̊, s̋, š, s̍, s̎, s̏, s̐, s̑, s̒, s̓, s̔, s̕, s̖, s̗, s̘, s̙, s̚, s̛, s̜, s̝, s̞, s̟, s̠, s̡, s̢, ṣ, s̤, s̥, ș, ş, s̨, s̩, s̪, s̫, s̬, s̭, s̮, s̯, s̰, s̱, s̲, s̳, s̴, s̵, s̶, s̷, s̸, s̹, s̺, s̻, s̼, s̽, s̾, s̿},  
 8128 s}, % Cyr  
 8129 t = {t̃, t̂, t̄, t̅, t̆, ṫ, ẗ, t̉, t̊, t̋, ť, t̍, t̎, t̏, t̐, t̑, t̒, t̓, t̔, t̕, t̖, t̗, t̘, t̙, t̚, t̛, t̜, t̝, t̞, t̟, t̠, t̡, t̢, ṭ, t̤, t̥, ț, ţ, t̨, t̩, t̪, t̫, t̬, ṱ, t̮, t̯, t̰, ṯ, t̲, t̳, t̴, t̵, t̶, t̷, t̸, t̹, t̺, t̻, t̼, t̽, t̾, t̿},  
 8130 u = {ù̃, ù̂, ù̄, ù̅, ù̆, ù̇, ù̈, ù̉, ù̊, ù̋, ù̌, ù̍, ù̎, ù̏, ù̐, ù̑, ù̒, ù̓, ù̔, ù̕, ù̖, ù̗, ù̘, ù̙, ù̚, ừ, ù̜, ù̝, ù̞, ù̟, ù̠, ù̡, ù̢, ụ̀, ṳ̀, ù̥, ù̦, ù̧, ų̀, ù̩, ù̪, ù̫, ù̬, ṷ̀, ù̮, ù̯, ṵ̀, ù̱, ù̲, ù̳, ù̴, ù̵, ù̶, ù̷, ù̸, ù̹, ù̺, ù̻, ù̼, ù̽, ù̾, ù̿},  
 8131 v = {ṽ, v̂},  
 8132 w = {ŵ̃, ŵ̂, ŵ̄, ŵ̅, ŵ̆, ŵ̇, ŵ̈, ŵ̉, ŵ̊, ŵ̋, ŵ̌, ŵ̍, ŵ̎, ŵ̏, ŵ̐, ŵ̑, ŵ̒, ŵ̓, ŵ̔, ŵ̕, ŵ̖, ŵ̗, ŵ̘, ŵ̙, ŵ̚, ŵ̛, ŵ̜, ŵ̝, ŵ̞, ŵ̟, ŵ̠, ŵ̡, ŵ̢, ẉ̂, ŵ̤, ŵ̥, ŵ̦, ŵ̧, ŵ̨, ŵ̩, ŵ̪, ŵ̫, ŵ̬, ŵ̭, ŵ̮, ŵ̯, ŵ̰, ŵ̱, ŵ̲, ŵ̳, ŵ̴, ŵ̵, ŵ̶, ŵ̷, ŵ̸, ŵ̹, ŵ̺, ŵ̻, ŵ̼, ŵ̽, ŵ̾, ŵ̿},  
 8133 w}, % Cyr  
 8134 x = {x̃, x̂, x̄},  
 8135 x, x̃}, % Cyr  
 8136 y = {ý̃, ý̂, ý̄, ý̅, ý̆, ý̇, ý̈, ý̉, ý̊, ý̋, ý̌, ý̍, ý̎, ý̏, ý̐, ý̑, ý̒, ý̓, ý̔, ý̕, ý̖, ý̗, ý̘, ý̙, ý̚, ý̛, ý̜, ý̝, ý̞, ý̟, ý̠, ý̡, ý̢, ỵ́, ý̤, ý̥, ý̦, ý̧, ý̨, ý̩, ý̪, ý̫, ý̬, ý̭, ý̮, ý̯, ý̰, ý̱, ý̲, ý̳, ý̴, ý̵, ý̶, ý̷, ý̸, ý̹, ý̺, ý̻, ý̼, ý̽, ý̾, ý̿},  
 8137 y, ý̃}, % Cyr  
 8138 z = {ž̃, ž̂, ž̄, ž̅, ž̆, ž̇, ž̈, ž̉, ž̊, ž̋, ž̌, ž̍, ž̎, ž̏, ž̐, ž̑, ž̒, ž̓, ž̔, ž̕, ž̖, ž̗, ž̘, ž̙, ž̚, ž̛, ž̜, ž̝, ž̞, ž̟, ž̠, ž̡, ž̢, ẓ̌, ž̤, ž̥, ž̦, ž̧, ž̨, ž̩, ž̪, ž̫, ž̬, ž̭, ž̮, ž̯, ž̰, ẕ̌, ž̲, ž̳, ž̴, ž̵, ž̶, ž̷, ž̸, ž̹, ž̺, ž̻, ž̼, ž̽, ž̾, ž̿},  
 8139 % Cyrillic  
 8140 Γ = {Γ̃, Γ̂, Γ̄, Γ̅, Γ̆, Γ̇, Γ̈, Γ̉, Γ̊, Γ̋, Γ̌, Γ̍, Γ̎, Γ̏, Γ̐, Γ̑, Γ̒, Γ̓, Γ̔, Γ̕, Γ̖, Γ̗, Γ̘, Γ̙, Γ̚, Γ̛, Γ̜, Γ̝, Γ̞, Γ̟, Γ̠, Γ̡, Γ̢, Γ̣, Γ̤, Γ̥, Γ̦, Γ̧, Γ̨, Γ̩, Γ̪, Γ̫, Γ̬, Γ̭, Γ̮, Γ̯, Γ̰, Γ̱, Γ̲, Γ̳, Γ̴, Γ̵, Γ̶, Γ̷, Γ̸, Γ̹, Γ̺, Γ̻, Γ̼, Γ̽, Γ̾, Γ̿},  
 8141 Ж = {Ж̃, Ж̂, Ж̄, Ж̅, Ӂ, Ж̇, Ӝ, Ж̉, Ж̊, Ж̋, Ж̌, Ж̍, Ж̎, Ж̏, Ж̐, Ж̑, Ж̒, Ж̓, Ж̔, Ж̕, Ж̖, Ж̗, Ж̘, Ж̙, Ж̚, Ж̛, Ж̜, Ж̝, Ж̞, Ж̟, Ж̠, Ж̡, Ж̢, Ж̣, Ж̤, Ж̥, Ж̦, Ж̧, Ж̨, Ж̩, Ж̪, Ж̫, Ж̬, Ж̭, Ж̮, Ж̯, Ж̰, Ж̱, Ж̲, Ж̳, Ж̴, Ж̵, Ж̶, Ж̷, Ж̸, Ж̹, Ж̺, Ж̻, Ж̼, Ж̽, Ж̾, Ж̿},  
 8142 З = {З̃, З̂, З̄, З̅, З̆, З̇, Ӟ, З̉, З̊, З̋, З̌, З̍, З̎, З̏, З̐, З̑, З̒, З̓, З̔, З̕, З̖, З̗, З̘, З̙, З̚, З̛, З̜, З̝, З̞, З̟, З̠, З̡, З̢, З̣, З̤, З̥, З̦, З̧, З̨, З̩, З̪, З̫, З̬, З̭, З̮, З̯, З̰, З̱, З̲, З̳, З̴, З̵, З̶, З̷, З̸, З̹, З̺, З̻, З̼, З̽, З̾, З̿},  
 8143 Л = {Л̃, Л̂, Л̄, Л̅, Л̆, Л̇, Л̈, Л̉, Л̊, Л̋, Л̌, Л̍, Л̎, Л̏, Л̐, Л̑, Л̒, Л̓, Л̔, Л̕, Л̖, Л̗, Л̘, Л̙, Л̚, Л̛, Л̜, Л̝, Л̞, Л̟, Л̠, Л̡, Л̢, Л̣, Л̤, Л̥, Л̦, Л̧, Л̨, Л̩, Л̪, Л̫, Л̬, Л̭, Л̮, Л̯, Л̰, Л̱, Л̲, Л̳, Л̴, Л̵, Л̶, Л̷, Л̸, Л̹, Л̺, Л̻, Л̼, Л̽, Л̾, Л̿},  
 8144 П = {П̃, П̂, П̄, П̅, П̆, П̇, П̈, П̉, П̊, П̋, П̌, П̍, П̎, П̏, П̐, П̑, П̒, П̓, П̔, П̕, П̖, П̗, П̘, П̙, П̚, П̛, П̜, П̝, П̞, П̟, П̠, П̡, П̢, П̣, П̤, П̥, П̦, П̧, П̨, П̩, П̪, П̫, П̬, П̭, П̮, П̯, П̰, П̱, П̲, П̳, П̴, П̵, П̶, П̷, П̸, П̹, П̺, П̻, П̼, П̽, П̾, П̿},  
 8145 У = {У̃, У̂, Ӯ, У̅, Ў, У̇, Ӱ, У̉, У̊, Ӳ, У̌, У̍, У̎, У̏, У̐, У̑, У̒, У̓, У̔, У̕, У̖, У̗, У̘, У̙, У̚, У̛, У̜, У̝, У̞, У̟, У̠, У̡, У̢, У̣, У̤, У̥, У̦, У̧, У̨, У̩, У̪, У̫, У̬, У̭, У̮, У̯, У̰, У̱, У̲, У̳, У̴, У̵, У̶, У̷, У̸, У̹, У̺, У̻, У̼, У̽, У̾, У̿},  
 8146 Ч = {Ч̃, Ч̂, Ч̄, Ч̅, Ч̆, Ч̇, Ӵ, Ч̉, Ч̊, Ч̋, Ч̌, Ч̍, Ч̎, Ч̏, Ч̐, Ч̑, Ч̒, Ч̓, Ч̔, Ч̕, Ч̖, Ч̗, Ч̘, Ч̙, Ч̚, Ч̛, Ч̜, Ч̝, Ч̞, Ч̟, Ч̠, Ч̡, Ч̢, Ч̣, Ч̤, Ч̥, Ч̦, Ч̧, Ч̨, Ч̩, Ч̪, Ч̫, Ч̬, Ч̭, Ч̮, Ч̯, Ч̰, Ч̱, Ч̲, Ч̳, Ч̴, Ч̵, Ч̶, Ч̷, Ч̸, Ч̹, Ч̺, Ч̻, Ч̼, Ч̽, Ч̾, Ч̿},  
 8147 Ъ = {Ъ̃, Ъ̂, Ъ̄, Ъ̅, Ъ̆, Ъ̇, Ъ̈, Ъ̉, Ъ̊, Ъ̋, Ъ̌, Ъ̍, Ъ̎, Ъ̏, Ъ̐, Ъ̑, Ъ̒, Ъ̓, Ъ̔, Ъ̕, Ъ̖, Ъ̗, Ъ̘, Ъ̙, Ъ̚, Ъ̛, Ъ̜, Ъ̝, Ъ̞, Ъ̟, Ъ̠, Ъ̡, Ъ̢, Ъ̣, Ъ̤, Ъ̥, Ъ̦, Ъ̧, Ъ̨, Ъ̩, Ъ̪, Ъ̫, Ъ̬, Ъ̭, Ъ̮, Ъ̯, Ъ̰, Ъ̱, Ъ̲, Ъ̳, Ъ̴, Ъ̵, Ъ̶, Ъ̷, Ъ̸, Ъ̹, Ъ̺, Ъ̻, Ъ̼, Ъ̽, Ъ̾, Ъ̿},  
 8148 Ы = {Ы̃, Ы̂, Ы̄, Ы̅, Ы̆, Ы̇, Ӹ, Ы̉, Ы̊, Ы̋, Ы̌, Ы̍, Ы̎, Ы̏, Ы̐, Ы̑, Ы̒, Ы̓, Ы̔, Ы̕, Ы̖, Ы̗, Ы̘, Ы̙, Ы̚, Ы̛, Ы̜, Ы̝, Ы̞, Ы̟, Ы̠, Ы̡, Ы̢, Ы̣, Ы̤, Ы̥, Ы̦, Ы̧, Ы̨, Ы̩, Ы̪, Ы̫, Ы̬, Ы̭, Ы̮, Ы̯, Ы̰, Ы̱, Ы̲, Ы̳, Ы̴, Ы̵, Ы̶, Ы̷, Ы̸, Ы̹, Ы̺, Ы̻, Ы̼, Ы̽, Ы̾, Ы̿},  
 8149 Ь = {Ь̃, Ь̂, Ь̄, Ь̅, Ь̆, Ь̇, Ь̈, Ь̉, Ь̊, Ь̋, Ь̌, Ь̍, Ь̎, Ь̏, Ь̐, Ь̑, Ь̒, Ь̓, Ь̔, Ь̕, Ь̖, Ь̗, Ь̘, Ь̙, Ь̚, Ь̛, Ь̜, Ь̝, Ь̞, Ь̟, Ь̠, Ь̡, Ь̢, Ь̣, Ь̤, Ь̥, Ь̦, Ь̧, Ь̨, Ь̩, Ь̪, Ь̫, Ь̬, Ь̭, Ь̮, Ь̯, Ь̰, Ь̱, Ь̲, Ь̳, Ь̴, Ь̵, Ь̶, Ь̷, Ь̸, Ь̹, Ь̺, Ь̻, Ь̼, Ь̽, Ь̾, Ь̿},  
 8150 Ҁ = {Ҁ̃, Ҁ̂, Ҁ̄, Ҁ̅, Ҁ̆, Ҁ̇, Ҁ̈, Ҁ̉, Ҁ̊, Ҁ̋, Ҁ̌, Ҁ̍, Ҁ̎, Ҁ̏, Ҁ̐, Ҁ̑, Ҁ̒, Ҁ̓, Ҁ̔, Ҁ̕, Ҁ̖, Ҁ̗, Ҁ̘, Ҁ̙, Ҁ̚, Ҁ̛, Ҁ̜, Ҁ̝, Ҁ̞, Ҁ̟, Ҁ̠, Ҁ̡, Ҁ̢, Ҁ̣, Ҁ̤, Ҁ̥, Ҁ̦, Ҁ̧, Ҁ̨, Ҁ̩, Ҁ̪, Ҁ̫, Ҁ̬, Ҁ̭, Ҁ̮, Ҁ̯, Ҁ̰, Ҁ̱, Ҁ̲, Ҁ̳, Ҁ̴, Ҁ̵, Ҁ̶, Ҁ̷, Ҁ̸, Ҁ̹, Ҁ̺, Ҁ̻, Ҁ̼, Ҁ̽, Ҁ̾, Ҁ̿},

```

8151 Ж = {ж, ђ, Ӓ},
8152 З = {з, ӑ},
8153 И = {й, й̃, й̂, й̄, й̆},
8154 К = {к, к̃, к̂, к̄, к̆, к̈},
8155 Л = {л},
8156 М = {м},
8157 Н = {н, н̃, н̂, н̄},
8158 П = {п},
8159 Т = {т},
8160 Х = {х, х̃},
8161 Ч = {ч, ч̃, ч̂, ч̄},
8162 Ш = {ш},
8163 Ы = {ы̃},
8164 Э = {э̃},
8165 Ё = {ё̃},
8166 Ә = {ә̃},
8167 У = {ұ̃},
8168 Γ = {Γ}, % Greek
8169 Π = {Π}, % Greek
8170 }
8171
8172 % missing: tipa, math, symbols, ...
8173 </CharisSIL>
8174 <*Palatino>
8175 \DeclareCharacterInheritance
8176 { encoding = {EU1,EU2,TU},
8177 family = {Palatino} }

```

Unfortunately, I don't have a Palatino variant containing all of the following glyphs. The settings are typeset in T<sub>E</sub>X 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`.

[illegible]

[illegible]8229 */Palatino*

## Less characters in the Lato font ...

```

8230 (*Lato)
8231 \DeclareCharacterInheritance
8232 { encoding = {TU,EU1,EU2},
8233   family = {Lato} }
8234
8235 { A = {Ä,Å,Ă,Ȧ,Ȧ,Ȧ,A},
8236   a = {ä,á,â,ã,ä,ä,a},
8237   C = {Ć,Ç},
8238   c = {ć,ç},
8239   D = {Ð},
8240   E = {Ê,Ê,Ê,Ê,Ê},
8241   e = {ê,ê,ê,ê,ê},
8242   I = {İ,İ,İ,İ},
8243   i = {ı,ı,ı,ı},
8244   L = {Ł},
8245   l = {ł},
8246   N = {Ñ,Ñ},
8247   n = {ñ,ñ},
8248   O = {Ø,Ó,Ó,Ô,Ö,Ö},
8249   o = {ø,ò,ó,ô,ö,ö},
8250   S = {Ś,Ś},
8251   s = {ś,ś},
8252   U = {Ü,Ü,Ü,Ü},
8253   u = {ü,ú,ú,ü},
8254   Y = {Ÿ,Ÿ},
8255   y = {ÿ,ÿ},
8256   Z = {Ž,Ž,Ž},
8257   z = {ž,ž,ž}
8258 }
```

8258 *</Lato>*

... and even less in the self-professedly awesome Font Awesome font.

```
8259 (*FontAwesome)
8260 \DeclareCharacterInheritance
8261 { encoding = {TU,EU1,EU2},
8262   family   = {FontAwesome} }
8263 { }
8264 (/FontAwesome)
```

## 16.2 Character protrusion

```

8265
8266 %%% -----
8267 %%% PROTRUSION
8268
8269 (*LatinModernRoman)
8270 \SetProtrusion
8271   [ name      = LMR-default ]
8272   { encoding = {EU1,EU2,TU},
8273     family    = Latin Modern Roman }
8274   {
8275     A = {50,50},
8276     Æ = {50, },
8277     F = { ,50},
8278     J = {50, },
8279     K = { ,50},
8280     L = { ,50},
8281     T = {50,50},
8282     V = {50,50},
8283     W = {50,50},
8284     X = {50,50},
8285     Y = {50,50},
8286     k = { ,50},
8287     r = { ,50},
8288     t = { ,70},
8289     v = {50,50},
8290     w = {50,50},
8291     x = {50,50},
8292     y = {50,70},
8293     0 = { ,50},
8294     1 = {100,200},
8295     2 = {50,50},
8296     3 = {50,50},
8297     4 = {70,70},
8298     5 = { ,50},
8299     6 = { ,50},
8300     7 = {50,100},
8301     8 = { ,50},
8302     9 = { ,50},
8303     . = { ,70},
8304     {,}= { ,500},
8305     := { ,500},
8306     ; = { ,500},
8307     != { ,100},
8308     ? = { ,200},
8309     @ = {50,50},
8310     ~ = {200,250},
8311     \% = {50,50},
8312     * = {300,300},
8313     + = {250,250},
8314     - = {400,500}, % /hyphen
8315     – = {400,300}, % /endash
8316     — = {300,200}, % /emdash
8317     _ = {200,200}, % /underscore
8318     / = {200,300},
8319     /backslash = {200,300},
8320     ' = {300,400}, % /quotesingle
8321     ‘ = {500,700}, ’ = {500,600},
8322     “ = {500,300}, ” = {200,600},
8323     , = {400,400}, „ = {400,400},
8324     ‹ = {400,400}, › = {300,500},
8325     « = {300,200}, » = {100,400},
8326     ¡ = {100, }, ¿ = {100, },
8327     (= {300, }, ) = { ,300},

```

```

8328    < = {200,100}, > = {100,200},
8329    /braceleft = {400,200}, /braceright = {200,400},
8330    /angleleft = {400,  }, /angleright = {  ,400},
8331    † = {100,100},
8332    ‡ = { 80, 80},
8333    • = {200,200},
8334    · = {400,450}, % / periodcentered
8335    °C = { 80, 50},
8336    ℄ = {  , 50},
8337    ° = {400,400},
8338    ™ = {100,200},
8339    © = {100,100},
8340    ® = {100,100},
8341    ª = {100,200},
8342    º = {100,200},
8343    ¹ = {200,250},
8344    º = { 50,100},
8345    º = { 50,100},
8346    ¬ = {200,  },
8347    − = {300,300},
8348    ± = {150,200},
8349    × = {150,250},
8350    ÷ = {150,250},
8351    € = {100,  },
8352    /one.oldstyle = {100,100},
8353    /two.oldstyle = { 50, 50},
8354    /three.oldstyle = { 30, 80},
8355    /four.oldstyle = { 50, 50},
8356    /seven.oldstyle = { 50, 80},
8357    Γ = {  ,180}, % /Gamma
8358    Δ = {100,100}, % /Delta
8359    Θ = { 50, 50}, % /Theta
8360    Λ = {100,100}, % /Lambda
8361    % Ξ = {,}, % /Xi
8362    % Π = {,}, % /Pi
8363    Σ = { 50, 50}, % /Sigma
8364    Υ = {100,100}, % /Upsilon
8365    Φ = { 50, 50}, % /Phi
8366    Ψ = { 50, 50}, % /Psi
8367    % Ω = {,}, % /Omega
8368    }
8369
8370 \SetProtrusion
8371 [ name = LMR-it ]
8372 { encoding = {EU1,EU2,TU},
8373   family = Latin Modern Roman,
8374   shape = {it,sl} }
8375 {
8376   A = {125,100},
8377   Æ = {125,-55},
8378   B = {90,-40},
8379   C = {145,-75},
8380   D = {75,-28},
8381   E = {80,-55},
8382   F = {85,-80},
8383   G = {153,-15},
8384   H = {73,-60},
8385   I = {140,-120},
8386   IJ = {140,-80},
8387   J = {135,-80},
8388   K = {70,-30},
8389   L = {87, 40},
8390   M = {67,-45},
8391   N = {75,-55},
8392   O = {150,-30},

```

```

8393    CE = {150,-55},
8394    P = {82,-50},
8395    Q = {150,-30},
8396    R = {75, 15},
8397    S = {90,-65},
8398    $ = {100,-20},
8399    T = {220,-85},
8400    U = {230,-55},
8401    V = {260,-60},
8402    W = {185,-55},
8403    X = {70,-30},
8404    Y = {250,-60},
8405    Z = {90,-60},
8406    a = {150,-10},
8407    b = {170, },
8408    c = {173,-10},
8409    d = {150,-55},
8410    e = {180, },
8411    f = { , -250},
8412    g = {150,-10},
8413    h = {100, },
8414    i = {210, },
8415    ij = {210,-40},
8416    j = { , -40},
8417    k = {110,-50},
8418    l = {240,-110},
8419    m = {80, },
8420    n = {115, },
8421    o = {155, },
8422    q = {170,-40},
8423    r = {155,-40},
8424    s = {130, },
8425    t = {230,-10},
8426    u = {120, },
8427    v = {140,-25},
8428    w = {98,-20},
8429    x = {65,-40},
8430    y = {130,-20},
8431    z = {110,-80},
8432    0 = {170,-85},
8433    1 = {230,110},
8434    2 = {130,-70},
8435    3 = {140,-70},
8436    4 = {130,80},
8437    5 = {160, },
8438    6 = {175,-30},
8439    7 = {250,-150},
8440    8 = {130,-40},
8441    9 = {155,-80},
8442    . = { ,500},
8443    {,}= { ,450},
8444    := { ,300},
8445    ; = { ,300},
8446    & = {130,30},
8447    \% = {180,50},
8448    * = {380,20},
8449    + = {180,200},
8450    @ = {180,10},
8451    ~ = {200,150},
8452    ( = {300, }, ) = { ,70},
8453    / = {100,100},
8454    - = {500,300}, % /hyphen
8455    - = {500,300}, % /endash
8456    — = {400,170}, % /emdash
8457    _ = {100,200}, % /underscore

```



```

8458 ' = {300,400}, % /quotesingle
8459 " = {500,300},
8460 ‘ = {800,200}, ’ = {800,-20},
8461 “ = {540,100}, ” = {500,100},
8462 , = {300,700}, „ = {200,600},
8463 ‹ = {500,300}, › = {400,400},
8464 « = {400,100}, » = {200,300},
8465 ¡ = {200, }, ¿ = {200, },
8466 < = {300,100}, > = {200,100},
8467 /backslash = {300,300},
8468 /braceleft = {400,100}, /braceright = {200,200},
8469 † = {200, 80},
8470 ‡ = {120, 80},
8471 • = {220,100},
8472 · = {550,300}, % / periodcentered
8473 °C = {170, },
8474 ¢ = {100, 50},
8475 ¶ = {200, },
8476 ° = {500,300},
8477 ™ = {200, 70},
8478 © = { 50, 70},
8479 ® = { 50, 70},
8480 º = {140,100},
8481 º = {140,100},
8482 ¹ = {400,150},
8483 º = {250, 80},
8484 º = {250, 80},
8485 ¬ = {250, 80},
8486 − = {300,200},
8487 ± = {150,170},
8488 × = {200,200},
8489 ÷ = {200,200},
8490 € = {150, },
8491 /one.oldstyle = {100,100},
8492 /two.oldstyle = {100, 80},
8493 /three.oldstyle = { 80, 50},
8494 /four.oldstyle = { 80, 80},
8495 /five.oldstyle = { 50, },
8496 /six.oldstyle = { 50, },
8497 /seven.oldstyle = { 80, 80},
8498 /eight.oldstyle = { 50, },
8499 Γ = {100,120}, % /Gamma
8500 Δ = {120,100}, % /Delta
8501 Θ = {120, 50}, % /Theta
8502 Λ = {130,100}, % /Lambda
8503 Ξ = {100,}, % /Xi
8504 Π = {100,}, % /Pi
8505 Σ = {100, 50}, % /Sigma
8506 Υ = {180,100}, % /Upsilon
8507 Φ = {130, 70}, % /Phi
8508 Ψ = {130, 50}, % /Psi
8509 Ω = { 50,}, % /Omega
8510 }
8511 </LatinModernRoman>
8512 <*CharisSIL>
8513 \SetProtrusion
8514 [ name = Charis-default ]
8515 { encoding = {EU1,EU2,TU},
8516 family = Charis SIL }
8517 {
8518 A = {50,50},
8519 Æ = {50,50},
8520 C = {50, },
8521 D = { ,50},
8522 F = { ,50},

```

```

8523 G = {50, },
8524 J = {100, },
8525 K = { ,50},
8526 L = { ,50},
8527 Ḷ = { ,100},
8528 O = {50,50},
8529 Œ = {50, },
8530 P = { ,50},
8531 Q = {50,70},
8532 R = { ,50},
8533 ß = { ,40}, % capital sharp s
8534 T = {50,50},
8535 V = {50,50},
8536 W = {50,50},
8537 X = {50,50},
8538 Y = {50,50},
8539 k = { ,50},
8540 ḷ = { ,150},
8541 r = { ,50},
8542 t = { ,50},
8543 v = {50,50},
8544 w = {50,50},
8545 x = {50,50},
8546 y = { ,50},
8547 1 = {150,150},
8548 2 = {50,50},
8549 3 = {50, },
8550 4 = {100,50},
8551 6 = {50, },
8552 7 = {50,80},
8553 9 = {50,50},
8554 . = { ,600},
8555 {,} = { ,500},
8556 : = { ,400},
8557 ; = { ,300},
8558 ! = { ,100},
8559 ? = { ,200},
8560 @ = {50,50},
8561 ~ = {200,250},
8562 \% = { ,50},
8563 * = {300,300},
8564 + = {200,250},
8565 / = { ,200},
8566 /backslash = {150,200},
8567 | = {200,200},
8568 - = {400,500}, % hyphen
8569 - = {200,300}, % endash
8570 — = {150,250}, % emdash
8571 — = {200,200}, % Horizontal Bar = \texttwelveudash
8572 - = {150,150}, % Figure Dash = \textthreequartersemdash
8573 _ = {100,100},
8574 {=} = {100,100},
8575 ‘ = {300,400}, ’ = {300,400},
8576 “ = {300,300}, ” = {300,300},
8577 , = {400,400}, „ = {300,300},
8578 ‹ = {400,300}, › = {300,400},
8579 « = {200,200}, » = {150,300},
8580 ¡ = {100, }, ¿ = {100, },
8581 ( = {200, }, ) = { ,200},
8582 < = {200,150}, > = {100,200},
8583 [ = {100, }, ] = { ,100},
8584 /braceleft = {200, }, /braceright = { ,300},
8585 † = { 80, 80},
8586 ‡ = {100,100},
8587 • = {200,200},

```

8588 ° = {150,200},  
8589 ™ = {150,150},  
8590 ¢ = { 50, },  
8591 £ = { 50, },  
8592 ¦ = {200,200},  
8593 © = {100,100},  
8594 ® = {100,100},  
8595 ª = {100,200},  
8596 º = {200,200},  
8597 ¬ = {200, 50},  
8598 µ = { ,100},  
8599 ¶ = { ,100},  
8600 · = {300,400},  
8601 <sup>1</sup> = {200,300},  
8602 <sup>2</sup> = {100,200},  
8603 <sup>3</sup> = {100,200},  
8604 € = {100, },  
8605 ± = {150,200},  
8606 × = {200,200},  
8607 ÷ = {250,250},  
8608 /minus = {200,200},  
8609 − = {200,200},  
8610 % Cyrillic  
8611 Б = { ,50},  
8612 Г = { ,130},  
8613 Ж = {50,50},  
8614 З = {30,50},  
8615 Л = {50, },  
8616 У = {50,50},  
8617 Ф = {50,50},  
8618 Ч = {100, },  
8619 Ъ = { ,50},  
8620 б = { ,50},  
8621 Э = {50,50},  
8622 Ю = { ,40},  
8623 Я = {50, },  
8624 В = {50,50},  
8625 Ё = {50, },  
8626 Ъ = {50,100},  
8627 Ѓ = {50, },  
8628 Љ = {50,50},  
8629 Њ = { ,50},  
8630 Ћ = {50,50},  
8631 Ќ = {100,100},  
8632 Ў = {50,50},  
8633 Ъ = { ,50},  
8634 Ь = { ,50},  
8635 Љ = {50,80},  
8636 Њ = { ,80},  
8637 Ћ = {50,50},  
8638 Ќ = {50, },  
8639 ЈХ = {50,40},  
8640 К = { ,50},  
8641 ЈЕ = {50, },  
8642 ЈЂ = { ,50},  
8643 ЈЃ = { ,50},  
8644 Ѓ = { ,100},  
8645 Ѓ = {50,50},  
8646 г = { ,70},  
8647 к = { ,50},  
8648 л = {50, },  
8649 т = {50,50},  
8650 ф = {50,50},  
8651 ч = {50, },  
8652 ъ = { ,50},

```

8653   Ъ = { ,50},
8654   Э = { ,50},
8655   Я = {50, },
8656   Ь = {50, },
8657   Њ = { ,50},
8658   Ћ = { ,50},
8659   V = {50,50},
8660   Ў = {50, },
8661   Ъ = { ,50},
8662   Y = {50,50},
8663   Ы = { ,50},
8664   Ь = { ,50},
8665   Ь = { ,100},
8666   Ъ = {100,100},
8667   ʒ = {50,50},
8668   Ь = {50,70},
8669   Њ = { ,70},
8670   ʒ = {50,30},
8671   Ь = { ,50},
8672   Ь = { ,50},
8673   %   Д П Ц Ш Щ Ы Ъ Ы Э Ц Ы Э Э д
8674   %   в д ж з и м н п ц ш ы ю ѓ є һ ц э є ц з д є ѓ л ж р
8675   % Greek
8676   Δ = {50,50},
8677   Ψ = {50,50},
8678   γ = {70,70},
8679   λ = {40,70},
8680   π = {40,50},
8681   ρ = { ,50},
8682   σ = { ,50},
8683   χ = {50,50},
8684 }
8685
8686 \SetProtrusion
8687   [ name      = Charis-it   ]
8688   { encoding = {EU1,EU2,TU},
8689     family   = Charis SIL,
8690     shape     = {it,sl} }
8691   {
8692   C = {50, },
8693   G = {50, },
8694   J = {50, },
8695   L = {50,50},
8696   O = {50, },
8697   Œ = {50, },
8698   Q = {50, },
8699   S = {50, },
8700   $ = {50, },
8701   T = {70, },
8702   o = {50,50},
8703   p = { ,50},
8704   q = {50, },
8705   t = { ,50},
8706   w = { ,50},
8707   y = { ,50},
8708   l = {150,100},
8709   3 = {50, },
8710   4 = {100, },
8711   6 = {50, },
8712   7 = {100, },
8713   . = { ,700},
8714   {,} = { ,600},
8715   : = { ,400},
8716   ; = { ,400},
8717   ? = { ,150},

```

```

8718   & = { ,80},
8719   \% = {50,50},
8720   * = {300,200},
8721   + = {250,250},
8722   @ = {80,50},
8723   ~ = {150,150},
8724   / = { ,150},
8725   /backslash = {150,150},
8726   - = {300,400}, % hyphen
8727   – = {200,300}, % endash
8728   — = {150,200}, % emdash
8729   _ = { ,100},
8730   {=} = {200,200},
8731   ± = {150,200},
8732   × = {250,250},
8733   ÷ = {250,250},
8734   ° = {150,200},
8735   · = {300,400},
8736   ‘ = {400,200}, ’ = {400,200},
8737   “ = {300,200}, ” = {400,200},
8738   , = {200,500}, „ = {150,500},
8739   ‹ = {300,400}, › = {200,500},
8740   « = {200,300}, » = {150,400},
8741   ( = {200, }, ) = { ,200},
8742   < = {200,200}, > = {200,200},
8743   /braceleft = {300, }, /braceright = { ,200},
8744   % Cyrillic
8745   Ж = {50,30},
8746   Л = {50, },
8747   У = {50,30},
8748   Ф = {50, },
8749   Ч = {100, },
8750   Ъ = { ,50},
8751   Ь = { ,50},
8752   Э = {50,50},
8753   Я = {50, },
8754   В = {50,50},
8755   Љ = {50,50},
8756   Њ = {140,100},
8757   Ѕ = {70,50},
8758   Ћ = {50,80},
8759   Ќ = { ,80},
8760   Ѝ = {50,50},
8761   Ў = {50,50},
8762   Д = {50,30},
8763   М = {50, },
8764   Ф = {50, },
8765   Ч = {50, },
8766   Ъ = { ,50},
8767   Ь = { ,50},
8768   Э = { ,50},
8769   Я = {50, },
8770   Љ = {50,50},
8771   Њ = { ,50},
8772   В = {50,50},
8773   Љ = { ,50},
8774   Њ = {140,100},
8775   Ѕ = {70,50},
8776   Ћ = {50,70},
8777   Ќ = { ,70},
8778   % Greek
8779   Γ = { ,130},
8780   Δ = {50,50},
8781   Ψ = {50,50},
8782   γ = {70,70},

```

```

8783   λ = {40,70},
8784   π = {40,50},
8785   ρ = { ,50},
8786   σ = { ,50},
8787   χ = {50,50},
8788   }

```

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 Lua<sub>T</sub><sub>E</sub>X (where we can simply query the font version) and with X<sub>Y</sub><sub>T</sub><sub>E</sub>X (where we check for glyph name).

```

8789
8790 % quick and dirty -- maybe we'll promote this to a
8791 % regular key some time
8792 \define@key{MT@pr@c}{command}{\csname #1\endcsname}
8793
8794 % glyph names have changed with version 5.0 of Charis SIL:
8795 % before: /a.SC, /b.SC, ...
8796 % after:  /a.sc, /b.sc, ...
8797 \ifx\MT@lua\undefined
8798   \gdef\MT@get@CHARIS@SC{
8799     % test whether glyph "a.sc" exists
8800     \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
8801       \gdef\MT@CHARIS@SC{sc}%
8802     \else
8803       \gdef\MT@CHARIS@SC{SC}%
8804     \fi
8805   }
8806 \else
8807   \gdef\MT@get@CHARIS@SC{
8808     \gdef\MT@CHARIS@SC{\MT@lua{
8809       % check font version
8810 % -- why doesn't this work?:
8811 %   f = font.getfont(font.current());
8812 %   i = fontloader.info(f.filename);
8813 %   if (tonumber(i.version) < 5) then;
8814 %     if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
8815 %       tex.print("SC");
8816 %     else;
8817 %       tex.print("sc");
8818 %     end
8819 %   }}
8820   }
8821 \fi
8822
8823 \SetProtrusion
8824 [ name      = Charis-sc,
8825   load      = Charis-default,
8826   command   = {MT@get@CHARIS@SC} ]
8827 { encoding = {EU1,EU2,TU},
8828   family   = Charis SIL,
8829   shape     = {sc} }
8830 {
8831 % A = {100,100}, % etc., doesn't work with \textsc
8832 /a.\MT@CHARIS@SC = {100,100},
8833 /c.\MT@CHARIS@SC = {50, },
8834 /d.\MT@CHARIS@SC = { ,50},
8835 /f.\MT@CHARIS@SC = { ,50},
8836 /g.\MT@CHARIS@SC = {50, },
8837 /j.\MT@CHARIS@SC = {100, },
8838 /k.\MT@CHARIS@SC = { ,50},
8839 /l.\MT@CHARIS@SC = { ,50},
8840 /f l.\MT@CHARIS@SC = { ,50},
8841 /o.\MT@CHARIS@SC = {50,50},
8842 /oe.\MT@CHARIS@SC = {50, },

```

```

8843 /q.\MT@CHARIS@SC = {50,70},
8844 /r.\MT@CHARIS@SC = { ,50},
8845 /t.\MT@CHARIS@SC = {50,100},
8846 /v.\MT@CHARIS@SC = {50,50},
8847 /w.\MT@CHARIS@SC = {50,50},
8848 /x.\MT@CHARIS@SC = {50,50},
8849 /y.\MT@CHARIS@SC = {50,50}
8850 }
8851 (/CharisSIL)
8852 (*Palatino)
8853 \SetProtrusion
8854 [ name = palatino-default ]
8855 { encoding = {EU1,EU2,TU},
8856   family = {Palatino} }
8857 {
8858 A = {50,50},
8859 D = { ,50},
8860 J = {50, },
8861 K = { ,50},
8862 L = { ,50},
8863 O = {25, },
8864 T = {50,50},
8865 V = {50,50},
8866 W = {50,50},
8867 X = {50,50},
8868 Y = {50,50},
8869 b = { ,25},
8870 d = {25,30},
8871 f = { ,50},
8872 g = { ,100},
8873 k = { ,50},
8874 p = { ,50},
8875 q = {50, },
8876 r = { ,50},
8877 t = { ,50}, ◆ = { ,50}, ◆ = { ,50},
8878 v = {75,50},
8879 w = {50,50},
8880 x = {50,50},
8881 y = {50,70},
8882 1 = {100,50},
8883 2 = {25,50},
8884 4 = {50, },
8885 6 = {50, },
8886 9 = {25, },
8887 Æ = {100, },
8888 Œ = {25, },
8889 . = { ,700}, .. = { ,350}, ... = { ,150},
8890 {,} = { ,500},
8891 : = { ,500},
8892 ; = { ,500},
8893 ! = { ,100}, !! = { ,100},
8894 ? = { ,200}, ? = { ,200},
8895 @ = {50,50},
8896 ~ = {200,250},
8897 & = {50,100},
8898 \% = {100,100},
8899 * = {200,200},
8900 + = {250,250},
8901 ( = {100, }, ) = { ,300},
8902 / = {200,300},
8903 - = {400,500},
8904 \textendash = {300,300}, \textemdash = {200,200},
8905 \textquoteleft = {500,700}, \textquoteright = {500,700},
8906 \textquotedblleft = {300,400}, \textquotedblright = {300,400},
8907 \textbackslash = {200,300},

```

```

8908 \quotesinglbase = {400,400}, \quotedblbase = {400,400},
8909 \guilsinglleft = {400,400}, \guilsinglright = {300,500},
8910 \guillemotleft = {300,300}, \guillemotright = {200,400},
8911 \textexclamdown = {100, }, \textquestiondown = {100, },
8912 \textbraceleft = {400,200}, \textbraceright = {200,400},
8913 \textless = {200,100}, \textgreater = {100,200},
8914 ≤ = {200,100}, ≥ = {100,200},
8915 \textminus = {300,300},
8916 \texttrademark = {200,200},
8917 \textcopyright = {200,200},
8918 \textregistered = {200,200},
8919 \textdegree = {300,300},
8920 ¡ = {450,500}, ¬ = {250,150},
8921 ¬ = {150,250},
8922 · = {850,700},
8923 ¶ = {100,0},
8924 × = {150,300},
8925 ª = {300,300}, º = {300,300},
8926 ° = {200,400},
8927 ¹ = {400,350}, º = {200,300}, ³ = {250,400},
8928 ⁴ = {250,350}, ⁵ = {200,300}, ⁶ = {250,400},
8929 ⁷ = {200,450}, ⁸ = {250,400}, ⁹ = {200,350},
8930 ⁰ = {200,400},
8931 ¹ = {400,250}, ² = {200,300}, ³ = {250,400},
8932 ⁴ = {250,350}, ⁵ = {200,300}, ⁶ = {250,400},
8933 ⁷ = {200,450}, ⁸ = {250,400}, ⁹ = {200,350},
8934 ± = {150,100}, ÷ = {300,300},
8935 þ = { ,25},
8936 ¸ = {300,450}, ¸ = {300,450},
8937 ¸ = {300,450}, ¸ = {300,450},
8938 † = {200,250}, ‡ = {200,250},
8939 π = {50, },
8940 f = { ,50},
8941 № = {100,150},
8942 \textservicemark = {100,200},
8943 - = {400,500}, - = {400,500}, - = {200,300},
8944 - = {205,305}, — = {200,300}, — = {50,150},
8945 • = {125,200},
8946 % /a.sc = {50,50},
8947 }
8948
8949 \SetProtrusion
8950 [ name = palatino-it ]
8951 { encoding = {EU1,EU2,TU},
8952 family = {Palatino},
8953 shape = {it,sl} }
8954 {
8955 A = {50,50},
8956 Æ = {50, },
8957 B = {50, },
8958 C = {50, },
8959 D = {50,50},
8960 E = {50, },
8961 F = {50, },
8962 G = {50, },
8963 H = {50, },
8964 K = {50, },
8965 L = {50, },
8966 O = {50, },
8967 Œ = {50, },
8968 P = {50, },
8969 Q = {50, },
8970 R = {50, },
8971 S = {50, },
8972 $ = {50, },

```



```

8973 T = {100, },
8974 U = {50, },
8975 V = {100,50},
8976 W = {50, },
8977 X = {50, },
8978 Y = {100,50},
8979 b = { ,50},
8980 c = {25, },
8981 g = {75, },
8982 i = {25, },
8983 m = { ,50},
8984 n = { ,50},
8985 p = { ,25},
8986 q = {25, },
8987 x = { ,50},
8988 1 = {100, },
8989 2 = {50, },
8990 4 = {50, },
8991 7 = {50, },
8992 . = { ,500}, .. = { ,350}, ... = { ,200},
8993 {,} = { ,500},
8994 : = { ,300},
8995 ; = { ,300},
8996 ? = { ,300},    ʔ = { ,300},
8997 & = {50,50},
8998 \% = {100,100},
8999 * = {200,200},
9000 + = {150,200},
9001 @ = {50,50},
9002 ~ = {200,150},
9003 ( = {200, }, ) = { ,200},
9004 / = {100,200},
9005 - = {300,500},
9006 \textendash = {300,300}, \textemdash = {200,200},
9007 \textquoteleft = {700,400}, \textquoteright = {700,400},
9008 \textquotedblleft = {500,300}, \textquotedblright = {500,300},
9009 _ = {100,100},
9010 \textbackslash = {100,200},
9011 \quotesinglbase = {500,500}, \quotedblbase = {400,400},
9012 \guilsinglleft = {400,400}, \guilsinglright = {300,500},
9013 \guillemotleft = {300,300}, \guillemotright = {300,300},
9014 \textexclamdown = {100, }, \textquestiondown = {200, },
9015 \textbraceleft = {200,100}, \textbraceright = {200,200},
9016 \textless = {300,100}, \textgreater = {200,100},
9017 ≤ = {200,100}, ≥ = {100,200},
9018 † = {450,500}, ‡ = {250,150},
9019 · = {850,700},
9020 ℥ = {100,0},
9021 × = {150,300},
9022 ° = {300,250}, ° = {300,300}, ° = {300,250},
9023 ° = {300,200},
9024 ¹ = {300,150}, º = {350,200}, º = {250,150},
9025 ² = {350,100}, ³ = {300,50}, ³ = {400,100},
9026 ⁴ = {400,50}, ⁵ = {250,50}, ⁵ = {300,50},
9027 ⁶ = {300,300},
9028 ⁷ = {300,350}, ⁸ = {300,150}, ⁹ = {250,250},
9029 ¹⁰ = {400,200}, ¹¹ = {300,100}, ¹² = {450,200},
9030 ¹³ = {450,150}, ¹⁴ = {400,250}, ¹⁵ = {400,200},
9031 ± = {150,100}, ÷ = {300,300},
9032 þ = {50, },
9033 † = {250,200}, ‡ = {250,200},
9034 . = {300,450}, . = {300,450},
9035 ¸ = {300,450}, ¸ = {300,450},
9036 - = {300,500}, - = {300,500}, - = {100,300},
9037 - = {125,305}, - = {200,300}, - = {125,150},

```

```
9038   • = {125,200}
9039   }
9040
9041 \SetProtrusion
9042   [ name      = palatino-sc,
9043     load      = palatino-default ]
9044   { encoding = {EU1,EU2,TU},
9045     family   = {Palatino},
9046     shape    = sc }
9047   {
9048     a = {50,50},
9049     æ = {50, },
9050     b = { 0,0},
9051     d = { 0,0},
9052     f = { 0,0},
9053     g = { 0,0},
9054     j = {50, },
9055     l = { ,50},
9056     o = { 0,0},
9057     p = { 0,0},
9058     q = { 0, },
9059     r = { , 0},
9060     t = {50,50},
9061     y = {50,50},
9062     fl = { 0,50},
9063     ffl = { 0,50},
9064     ◈ = { 0,50},
9065     ◈ = { 0,50}
9066   }
9067 </Palatino>
9068 <Lato> %% No settings yet.
9069 <*FontAwesome>
9070 \SetProtrusion
9071   [ name = empty ]
9072   { encoding = {TU,EU1,EU2},
9073     family   = {FontAwesome} }
9074   { }
9075 </FontAwesome>
9076
```

## 17 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```

9077 <test>
9078 \documentclass{article}
9079
9080 %% Here you can specify the font you want to test, using
9081 %% the commands \fontfamily, \fontseries and \fontshape.
9082 %% Make sure to end all lines with a comment character!
9083 \newcommand*\TestFont{%
9084   \fontfamily{ppl}%
9085   \fontseries{b}%
9086   \fontshape{it}% sc, sl
9087 }
9088
9089 \usepackage{ifthen}
9090 \usepackage[T1]{fontenc}
9091 \usepackage[latin1]{inputenc}
9092 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
9093
9094 \pagestyle{empty}
9095 \setlength{\parindent}{0pt}
9096 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
9097 \newcommand*\testprotrusion[2][\crulefill]{%
9098   \ifthenelse{equal{#1}{r}}{\crulefill}{\leftarrowfill} #2%
9099   lorem ipsum dolor sit amet,
9100   \ifthenelse{equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
9101   \ifthenelse{equal{#1}{l}}{\crulefill}{\rightarrowfill}
9102   you know the rest%
9103   \ifthenelse{equal{#1}{l}}{\crulefill}{\rightarrowfill} #2%
9104   \linebreak
9105   {\fontencoding{\encodingdefault}%
9106    \fontseries{\seriesdefault}%
9107    \fontshape{\shapedefault}%
9108    \selectfont
9109    Here is the beginning of a line, \dotfill and here is its end}\linebreak
9110 }
9111 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
9112 \def\stripprefix#1>{}
9113 \newcount\charcount
9114 \begin{document}
9115
9116 \microtypesetup{expansion=false}
9117
9118 {\centering The font in this document is called by:\\
9119 \texttt{\showTestFont}\par}\bigskip
9120
9121 \TestFont\selectfont
9122 This line intentionally left empty\linebreak
9123 %% A -- Z
9124 \charcount=65
9125 \loop
9126   \testprotrusion{\char\charcount}
9127   \advance\charcount 1
9128   \ifnum\charcount < 91 \repeat
9129 %% a -- z
9130 \charcount=97
9131 \loop
9132   \testprotrusion{\char\charcount}
9133   \advance\charcount 1
9134   \ifnum\charcount < 123 \repeat
9135 %% 0 -- 9
9136 \charcount=48
9137 \loop

```

```

9138 \testprotrusion{\char\charcount}
9139 \advance\charcount 1
9140 \ifnum\charcount < 58 \repeat
9141 %%
9142 \testprotrusion[r]{,}
9143 \testprotrusion[r]{.}
9144 \testprotrusion[r]{;}
9145 \testprotrusion[r]{:}
9146 \testprotrusion[r]{?}
9147 \testprotrusion[r]{!}
9148 \testprotrusion[l]{\textexclamdown}
9149 \testprotrusion[l]{\textquestiondown}
9150 \testprotrusion[r]{}}
9151 \testprotrusion[l]{(}
9152 \testprotrusion{/}
9153 \testprotrusion{\char~\\}
9154 \testprotrusion{-}
9155 \testprotrusion{\textendash}
9156 \testprotrusion{\textemdash}
9157 \testprotrusion{\textquoteleft}
9158 \testprotrusion{\textquoteright}
9159 \testprotrusion{\textquotedblleft}
9160 \testprotrusion{\textquotedblright}
9161 \testprotrusion{\quotesinglbase}
9162 \testprotrusion{\quotedblbase}
9163 \testprotrusion{\guilsinglleft}
9164 \testprotrusion{\guilsinglright}
9165 \testprotrusion{\guillemotleft}
9166 \testprotrusion{\guillemotright}
9167
9168 \newpage
9169 The following displays the current font stretched by 5%,
9170 normal, and shrunk by 5%:
9171
9172 \bigskip
9173 \newlength{\MTln}
9174 \newcommand*{\teststring}
9175 {ABCDEFGH IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
9176 \settowidth{\MTln}{\teststring}
9177 \microtypesetup{expansion=true}
9178
9179 \parbox{1.05\MTln}{\teststring\linebreak\\
9180 \teststring}\par\bigskip
9181 \parbox{0.95\MTln}{\teststring}
9182
9183 \end{document}
9184 /test

```

Needless to say that things may always be improved. For suggestions, mail to [w.m.l@gmx.net](mailto:w.m.l@gmx.net).

## 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}
```

9185 *(`*logo`)*

Here's how the logo on the title page was created.<sup>30</sup> 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.<sup>31</sup> 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`.

9186 `\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.

9187 `\input bbox.sty`

`\tempdim` Allocate some `dimen` registers.

9188 `\newdimen\tempdim`

`\fboxrulei` Frame width of the box as  $\TeX$  sees it.

9189 `\newdimen\fboxrulei`

9190 `\fboxrulei=0.1pt`

`\fboxruleii` Frame width of the bounding box.

9191 `\newdimen\fboxruleii`

9192 `\fboxruleii=0.1pt`

`\kernboxheight` Height of the box indicating the kern.

9193 `\newdimen\kernboxheight`

9194 `\kernboxheight=5pt`

`\scaletoem` An auxiliary macro. Return a dimension relative to the `em`-width of the font. Requires `e-TeX`.

9195 `\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).

9196 `\fontinstcc`

9197 `\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.

9198 `\ifdim\fontdimen6\font = 0pt`

9199 `\typeout{***-Warning:-no-fontdimen-6-specified-***^^J%}`

9200 `***-setting-it-to-\pdffontsize\font \ifnum\pdfptextversion < 130 pt\fi-***}`

9201 `\fontdimen6\font=\pdffontsize\font \ifnum\pdfptextversion < 130 pt\fi\relax`

9202 `\fi`

9203 `\installfonts`

9204 `\input_metrics{{\logofont,\metrics\printbbs{#1}}\relax}`

30 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.

31 Message ID: 42aa3687\$0\$24366\$9b4e6d93@newsread2.arcor-online.net

```

9205 \endinstallfonts
9206 }
9207 \normalcc
      Layers.
9208 \makeatletter
9209 \def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
9210 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9211 \ifx\mt@order\undefined\let\mt@order\@empty\fi
9212 \xdef\mt@order{\mt@order[(Logo)]}
9213 \let\mtl@resources\@empty
9214 \def\mtl@register#1{%
9215   \immediate\pdfobj{<< /Type/OCG /Name{#1} >>}
9216   \expandafter\xdef\csname mtl@#1\endcsname{\the\pdfobj\space 0 R }
9217   \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
9218   \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
9219   \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
9220 \mtl@register{canvas}
9221 \mtl@register{characters}
9222 \mtl@register{bounding-boxes}
9223 \mtl@register{TeX-boxes}
9224 \xdef\mt@order{\mt@order]}
9225 \global\let\mtl@objects\mt@objects
9226 \def\togglelayer#1#2{%
9227   \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
9228   user{/Subtype/Link
9229     /BS << /Type/Border/W 0 >> /H/0
9230     /A << /S/SetOCGState
9231     /State[/Toggle \csname mtl@#1\endcsname] >>
9232   }#2\pdfendlink
9233 }

```

\printbbs Preparation.

```

9234 \setcommand\printbbs#1{%
9235   \setbox0\hbox{#1}%
9236   \leavevmode
9237   \kern-\fboxrulei

```

The canvas in the natural width of the text minus protrusion, in color bgcolor.

```

9238 \mtl@layer{canvas}{%
9239   \getboundarychars#1\relax
9240   \tempdim=\dimexpr\wd0 - (\scaletoe{\lcode\font\firstchar}+
9241     \scaletoe{\rcode\font\lastchar})\relax
9242   \kern\dimexpr\scaletoe{\lcode\font\firstchar}\relax
9243   \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
9244     \hrule width \tempdim
9245     height \dimexpr\dp0+\ht0+0.15em\relax}%
9246   \kern-\tempdim

```

The baseline, in color blcolor.

```

9247   \vbox{\color{blcolor}%
9248     \hrule width \tempdim
9249     height \fboxrulei}%
9250   }%
9251   \kern-\dimexpr\wd0 -\scaletoe{\rcode\font\lastchar}\relax

```

The string.

```

9252 \printbbs #1\relax\relax
9253 }

```

\getboundarychars Get first ....

```

9254 \def\getboundarychars#1#2\relax{%
9255   \def\firstchar{`#1}%
9256   \getlastchar#1#2\relax
9257 }

```

\getlastchar ... and last character.

```

9258 \def\getlastchar#1#2{%

```

```

9259 \ifx\relax#2\relax
9260 \def\lastchar{`#1}%
9261 \else
9262 \expandafter\getlastchar
9263 \fi #2%
9264 }

```

`\printbbs` Loop over all characters of the string.

```

9265 \def\printbbs#1#2#3\relax{%
9266 \ifx\relax#1\relax
9267 \else
9268 \ifx\relax#2\relax
9269 \printbb{#1}{}%
9270 \else
9271 \printbb{#1}{#2}%
9272 \fi
9273 \expandafter\printbbs
9274 \fi #2#3\relax
9275 }

```

`\printbb` Record the kern between the current and the following character, then print the character. `\kerning` is a fontinst command.

```

9276 \setcommand\printbb#1#2{%
9277 \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
9278 \showboxes{#1}%

```

This could be another application.

```

9279 % \quad
9280 % w: \the\scaletom{\width{#1}},
9281 % bb: \the\scaletom{\bbleft{#1}}/%
9282 % \the\scaletom{\bbright{#1}},
9283 % \the\scaletom{\number\numexpr\width{#1}-\bbright{#1}\relax}
9284 % h: \height{#1}/\bbtop{#1}, \bbbotttom{#1}/\depth{#1}\par
9285 }

```

`\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'  $\neq$  ',').

```

9286 \setcommand\showboxes#1{%
9287 \leavevmode
9288 \color{texcolor}%

```

We have to record the width of the glyph.

```

9289 \setbox0\hbox{\color{textcolor}{#1}}%
9290 \global\tempdim=\wd0\relax
9291 \kern-\fboxrulei

```

1. *The  $\TeX$  box*: Print a frame in color `texcolor`. This frame shows the glyph as  $\TeX$  sees it.

```

9292 \mtl@layer{TeX-boxes}{%
9293 \hbox{%
9294 \lower\dimexpr \dp0 + \fboxrulei\relax
9295 \hbox{%
9296 \vbox{%
9297 \hrule height\fboxrulei
9298 \hbox{%
9299 \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
9300 \phantom{\unhcopy0}%
9301 \vrule width\fboxrulei
9302 }%
9303 \hrule height\fboxrulei}}}
9304 }%

```

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.

```

9305 \kern-\wd0
9306 \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.

```

9307 \kern\dimexpr\scaletom{\bbleft{#1}}-\tempdim-\fboxruleii\relax

```

3. *The bounding box*: will be printed in color `bbcolor`.

```

9308 \mtl@layer{bounding-boxes}{%
9309   {\color{bbcolor}%
9310     \hbox{%
9311       \lower\dimexpr-\scaletoe{\bbbottom{#1}}+\fboxruleii\relax
9312       \hbox{%
9313         \vbox{%
9314           \hrule height\fboxruleii
9315           \hbox to \dimexpr\scaletoe{\numexpr
9316             \bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{%
9317             \vrule height \dimexpr\scaletoe{\numexpr
9318               \bbtop{#1}-\bbbottom{#1}\relax}%
9319               width\fboxruleii
9320             \hfill
9321             \vrule width\fboxruleii}%
9322             \hrule height\fboxruleii}}}%
9323     }%
9324     \kern-\dimexpr\fboxruleii+\fboxrulei\relax
9325   }%

```

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.

```

9326 \kern\scaletoe{\numexpr\width{#1}-\bbright{#1}\relax}%
9327 \mtl@layer{TeX-boxes}{%
9328   {\ifnum\thekern<0
9329     \color{kerncolor}%
9330     \kern\scaletoe{\thekern}%
9331     \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoe{\thekern}\relax
9332       height \kernboxheight}%
9333     \kern\scaletoe{\thekern}%
9334   \else
9335     \color{texcolor}%
9336     \ifnum\thekern=0 \else
9337       \lower\kernboxheight
9338       \hbox{%
9339         \vbox{%
9340           \hrule height\fboxrulei
9341           \hbox{%
9342             \vrule height \kernboxheight width\fboxrulei
9343             \kern\dimexpr\scaletoe{\thekern}-2\fboxrulei\relax
9344             \vrule width\fboxrulei
9345           }%
9346           \hrule height\fboxruleii}}%
9347         \fi
9348       \fi
9349     }%
9350   }%
9351   % \kern-\fboxrulei
9352 }

```

```

9353 \newbox\logobox
9354 \def\printlogo{%
9355   \setbox\logobox=\hbox{\vbox{%
9356     \MakePercentComment

```

This is the Kepler MM font used in the logo.

```

9357 \def\logofont{pkpri9e10}
9358 \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
9359 \font\thellogofont=\logofont\space at 82pt

```

This would load the italic Palatino font instead.

```

9360 %\def\logofont{pplri}
9361 %\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
9362 %\edef\logofont{\logofont8r}
9363 %\font\thellogofont=\logofont\space at 78pt

```

Load the font.



```

9364 \thelogo font
      Protrusion values (overdone for didactic reasons).
9365 \lcode\font`M=96
9366 \rcode\font`e=46
      Now we can generate the logo.
9367 \pdfliteral direct{/SXS gs}%
9368 \showlogo{Microtype}%
9369 % \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
9370 % \kern5pt\[\[3\baselineskip]
9371 % \long\def\@makefnmark##1{%
9372 % \leftskip 0pt
9373 % \parindent 0pt
9374 % \everypar{\parindent 0pt}%
9375 % \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
9376 % \footnotetext[1]{This graphic display on a
9377 % \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
9378 % their \togglelayer{bounding-boxes}{bounding boxes}
9379 % and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
9380 }%
9381 \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
9382 \immediate\pdfobjj<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>%
9383 \immediate\pdfxform
9384 attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
9385 resources {/Properties <<\mtl@resources>>
9386 /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
9387 \logobox
9388 % \vskip-2.5\baselineskip
9389 % \leavevmode
9390 % \togglelayer{characters}{%
9391 % \pdfrefxform\pdflastxform
9392 % }%
9393 \pdfannot\logodimens{%
9394 /Subtype/Widget /FT/Btn /T(Logo)
9395 %/F 4 % why did I say this?
9396 /AP << /N \the\pdflastxform\space 0 R >>
9397 /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9398 /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9399 /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
9400 /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
9401 >> }%
9402 \vspace{3\baselineskip}
9403 }
      Our font.
9404 \pdfmapline{+pkpmmri8r10 Kep1MM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
      Define colours (thered and thegreen are copied from microtype.dtx).
9405 \def\mtdefinecolors{
9406 \definecolor{thered}{rgb}{0.65,0.04,0.07}
9407 \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
9408 \colorlet{texcolor}{thegreen!50} % TeX boxes
9409 \colorlet{kerncolor}{texcolor} % negative kerns
9410 \colorlet{bbcolor}{thered!50} % bounding box
9411 \colorlet{bgcolor}{black!8} % canvas
9412 \colorlet{blcolor}{black!50} % baseline
9413 \colorlet{textcolor}{black!40} % text
9414 }
      Use with microtype.dtx
9415 \ifx\documentclass\@twoclasseserror
9416 \usepackage{xcdraw}{xcolor}
9417 \mtdefinecolors
9418 \else

```

## A.2 Document

Now we can start the document.

```

9419 \documentclass[10pt,a4paper]{ltxdoc}
9420 \providecommand\MakePercentComment{\relax}
9421 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

    Re-use the preamble from microtype.dtx.
9422 \usepackage{microtype-doc}
9423 \usepackage{attachfile}
9424 \makeatletter
9425 \pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
9426 \makeatother
9427 \begin{document}

    You are currently reading this.
9428 \DocInput{microtype-logo.dtx}
9429 \newpage
9430 And here it is:
9431 \vfill
9432 \begin{center}
9433 \printlogo \null
9434 \end{center}
9435 \vfill
9436 \expandafter\enddocument
9437 \fi

    That's it.
9438 /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}
```

```

9439 \ifx\lssample\undefined
9440 *lssample
```

Upon popular request, here's how I've created the letterspacing illustration.<sup>32</sup>

### B.1 Macros

Rule width and image height and depth.

```

9441 \makeatletter
9442 \newdimen\lsamount
9443 \newdimen\lsrule
9444 \lsrule=0.2pt
9445 \def\lsheight{8pt}
9446 \def\lsdepth{12pt}
```

<sup>32</sup> 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).

```
9447 \def\lsfont{\fontfamily{paca}\selectfont}
      Loop over all letters in <#2>, letterspacing them by <#1>.
9448 \def\dols#1#2{\lsamount=#1\relax \dolss#2\enddols}
9449 \def\dolss#1#2\enddols{%
9450   \ifx\empty#2\empty\divide\lsamount 2\fi
9451   \ls{#1}%
9452   \ifx\empty#2\empty\else \dolss#2\enddols \fi
9453 }
```

One tikz picture for each letter.

```
9454 \def\ls#1{%
9455   \begin{tikzpicture}[remember picture,line width=\lsrule]
9456     \tikzstyle{every node}=[inner sep=0pt]
```

The bounding box.

```
9457     \mts@layer{stuff}{%
9458       \node[draw=thegrey,
9459         fill=theshade,
9460         outer sep=\lsrule,
9461         anchor=base,
9462         font=\lsfont]{\phantom{#1}};
9463     }
```

The letter.

```
9464     \node[anchor=base,font=\lsfont](#1){#1};
```

Two auxiliary coordinates.

```
9465     \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
9466     \path (#1.base east) ++(-.5\lsrule,-\lsdepth) coordinate (#1R);
9467     \mts@layer{stuff}{%
```

Now draw the normal character width,

```
9468       \draw[color=thered!75,
9469         fill=thered!30,
9470         outer sep=\lsrule]
9471         (#1L) rectangle (#1R);
9472       \ifdim\lsamount>0pt
9473         \path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
9474         \path (#1R) ++(\lsamount+\lsrule,\lsdepth) coordinate (#1E);
```

and the letter space.

```
9475       \draw[color=thered,
9476         fill=thered!50,
9477         outer sep=\lsrule]
9478         (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
9479     \fi
9480   }
9481 \end{tikzpicture}%
9482 \ignorespaces
9483 }
```

Draw the interword space.

```
9484 \def\lssp#1#2#3#4{%
9485   \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
9486     \mts@layer{stuff}{%
9487       \tikzstyle{every draw}=[anchor=bottom]
9488       \coordinate(#1space) at (#2/2,\lsdepth/2);
9489       \coordinate(#1stretch) at (#2+#3/2,+0pt);
9490       \coordinate(#1shrink) at (#2-#4/2,+0pt);
9491       \draw[color=thegreen,fill=thegreen!50,use as bounding box]
9492         (0,0) rectangle ++(#2,\lsdepth);
9493       \draw[color=thegreen,fill=thegreen!30]
9494         (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
9495       \draw[color=thegreen,fill=thegreen!50]
9496         (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
9497       \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
```

```

9498      (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
9499      \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
9500      (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
9501  }%
9502  \end{tikzpicture}%
9503  \ignorespaces
9504  }

Layers.
9505  \def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
9506  \def\mts@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
9507  \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9508  \ifx\mt@order\undefined\let\mt@order\@empty\fi
9509  \xdef\mt@order{\mt@order[(Sheep)]}
9510  \let\mts@resources\@empty
9511  \def\mts@register#1{%
9512    \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
9513    \expandafter\xdef\csname mts@#1\endcsname{\the\pdfastobj\space 0 R }
9514    \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
9515    \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
9516    \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
9517  \mts@register{stuff}
9518  \mts@register{tracking}
9519  \mts@register{ispace}
9520  \mts@register{ospace}
9521  \mts@register{istretch}
9522  \mts@register{ishrink}
9523  \mts@register{ostretch}
9524  \mts@register{oshrink}
9525  \mts@register{okern}
9526  \mts@register{ligature}
9527  \mts@register{_compatibility}
9528  \xdef\mt@order{\mt@order]}

Anchor point for the arrow in the code.
9529  \newcommand\anchorarrow[1]{%
9530    \tikz[remember picture,overlay]\node(#1_c){};}

Add an arrow from code to image.
9531  \newcommand\add@arrow[5][left]{%
9532    \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
9533      \mts@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
9534  }

Toggle layer.
9535  \def\toggle@layer#1#2#3{%
9536    \pdfstartlink
9537    user{/Subtype/Link
9538      /BS << /Type/Border/W 0 >> /H/0
9539    % /BS << /Type/Border/W 1 /S/D /D[4 1] >>
9540    % /C[0.7 0.7 0.7] /H/0
9541    /Contents(Click to Toggle!)
9542    /A << /S/SetOCGState
9543      /State[/Toggle \csname mts@#1\endcsname] >> }%
9544    \rlap{#2}%
9545    {\fboxsep=0pt \fboxrule=0pt
9546      \mts@layer{stuff}{%
9547        \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
9548        \mts@layer{#1}{%
9549          \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}}%
9550    }%
9551    \pdfendlink
9552  }
9553  \newcommand\showarrow[2][]{%
9554    \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
9555    \toggle@layer{\@tempa}{\itshape #2}}

```

The environment for our illustration.

```

9556 \def\ls@sample#1{%
9557   \parskip 4pt \parindent 0pt
9558   \par
9559   \vskip4pt
9560   {\leftskip 15pt
9561     \mt@pseudo@margin{\color{theblue}Click on the image to show the kerns
9562       and spacings involved. Click on emphasised words in the text below
9563       to reveal the relation of image and code.\strut}
9564     \mt@layer{_compatibility}%
9565     \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
9566       \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
9567     \mt@pseudo@margin{\color{thered}%
9568       If you had a \acronym{PDF} viewer that understands
9569       \acronym{PDF}\,,{\smaller1.5}, you could hide the arrows selectively.}}
9570     \vskip-\mt@unvdimen}%
9571   \vskip-4pt
9572   \setlength\fbboxsep{4pt}%
9573   \leavevmode
9574   \pdfstartlink
9575     user{/Subtype/Link
9576       /BS << /Type/Border/W 0 >> /H/0
9577       /A << /S/SetOCGState
9578         /State[/Toggle \mts@stuff] >> }%
9579     \fcolorbox{theframe}{theshade}%
9580     {\fontsize{34}{38}\selectfont #1}%
9581   \pdfendlink
9582   \par\medskip
9583   }%
9584   \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
9585 }

```

Now define the illustration to be used in the document.

```

9586 \def\lssample{%
9587   \ls@sample{%
9588     \dols{0pt}{Stop}
9589     \lssp{o}{0.45em}{0.25em}{0.15em}
9590     \dols{0.16em}{\stealing}\hskip-\dimexpr 0.08em+\lsrule\relax
9591     \lssp{i}{13.82pt}{4.65pt}{2.08pt}
9592     \dols{0.16em}{sheep}
9593     \dols{0pt}{!}
9594   }%

```

Don't forget to add the arrows.

```

9595   \vspace{-\baselineskip}
9596   \add@arrow{red}      {tracking}{lsamount_c.east}{a_ls}
9597   \add@arrow{red}      {okern}   {okernend_c.east}{p_ls}
9598   \add@arrow{green}    {ospace}  {ospace_c.east} {ospace}
9599   \add@arrow{green}    {ispace}  {ispace_c.center}{ispace}
9600   \add@arrow{green!75} {istretch}{istretch_c.east}{istretch.north}
9601   \add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
9602   \add@arrow{green!75} {ostretch}{ostretch_c.east}{ostretch.north}
9603   \add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
9604   \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
9605 }
9606 \fi

```

This is for use with microtype.dtx

```

9607 \ifx\documentclass\@twoclasseserror
9608   \usepackage{tikz}
9609 \else

```

## B.2 Document

```

9610 \documentclass[10pt,a4paper]{ltxdoc}
9611 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

```

Re-use the preamble from microtype.dtx.

```

9612 \usepackage{microtype-doc}
9613 \usepackage{attachfile}
9614 \usepackage{tikz}
9615 \makeatletter
9616 \pdfcatalog{/OCProperties << /OCGs [\mt@objects]
9617                               /D << /Order [\mt@order] /BaseState/OFF >> >> }
9618 \makeatother
9619 \begin{document}
  You are currently reading this.
9620 \DocInput{microtype-lssample.dtx}
  Now show what we are able to do.
9621 \noindent
9622 Since a picture is worth a thousand words, probably even more if, in our
9623 case, it depicts a couple of letterspaced words, let's bring one to sum up
9624 these somewhat confusing options. Suppose you had the following settings
9625 (which I would in no way recommend; they are only for illustrative purposes):
9626 \begin{verbatim}
9627 \SetTracking
9628   [ no ligatures = {"\anchorarrow{nolig}"f},
9629     spacing      = {60"\anchorarrow{ispace}"0*, "%
9630                    "-1"\anchorarrow{istretch}"00*, "\anchorarrow{ishrink}"},
9631     outer spacing = {4"\anchorarrow{ospace}"50, "%
9632                    "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
9633     outer kerning = {"\anchorarrow{okernbegin}"*, "%
9634                    "\anchorarrow{okernend}"*} ]
9635   { encoding = * }
9636   { 1"\anchorarrow{lsamount}"60 }
9637 \end{verbatim}
9638 and then write:
9639 \begin{verbatim}
9640 Stop \textls{stealing sheep}!
9641 \end{verbatim}
9642 this is the (typographically dubious) outcome:
9643
9644 \lssample
9645
9646 \noindent
9647 While the word 'Stop' is not letterspaced, the space between the letters in
9648 the other two words is expanded by the \showarrow[tracking]{tracking-amount}{red}
9649 of 160/1000\,em\,=\allowbreak\,0.16\,em.
9650 The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
9651 increased by 60\%, while its \showarrow[istretch]{stretch}{green} amount is
9652 decreased by 10\% and the \showarrow[ishrink]{shrink}{green} amount is left
9653 untouched.
9654 The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
9655 piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
9656 \showarrow[oshrink]{shrink}{green} by 0.15\,em.
9657 Note that there is no outer space after the text, since the exclamation mark
9658 immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
9659 of half the letterspace amount (0.08\,em) is added.
9660 Furthermore, one \showarrow{ligature}{grey} wasn't broken up, because we
9661 neglected to specify the '~|s|' in the |no ligatures| key.
9662
9663 \expandafter\enddocument
9664 \fi
9665 \lssample

```

## C Change history

### 2004/09/11 Version 1.0

General: Initial version . . . . . 1

### 2004/09/21 Version 1.1

General: configuration file names in lowercase (suggested by <i>Harald Harders</i> ) . . . . .	87	list . . . . .	89
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) . . . . .	147	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i> ) . . . . .	45
Protrusion: add factors for some more characters . . . . .	154	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . . . . .	94
settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . . . .	155	<code>\MT@pdfTeX@no</code> : fix: version check (reported by <i>Harald Harders</i> ) . . . . .	40
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance . . . . .	119	<code>\MT@permute</code> : don't use sets for empty encoding . . . . .	121
<code>\MT@declare@sets</code> : remove spaces around set name . . . . .	105	<code>\MT@setup@expansion</code> : issue an error instead of a warning, when pdfTeX version is too old for <code>autoexpand</code> . . . . .	136
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded . . . . .	87	<code>\MT@split@codes</code> : fix: allow zero and negative values . . . . .	63
<code>\MT@get@basefamily</code> : only remove suffixes 'x' or 'j' . . . . .	88	<code>\MT@use@set</code> : remove spaces around set name . . . . .	110
<code>\MT@get@listname@</code> : don't check for empty attributes			

### 2004/10/03 Version 1.2

Font aliases: declare <code>cmor</code> as an alias of <code>cmr</code> . . . . .	144	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to <code>\@empty</code> . . . . .	91
Font sets: new: <code>allmath</code> and <code>basicmath</code> . . . . .	143	<code>\MT@get@listname@</code> : alternatively check for alias font name . . . . .	89
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding . . . . .	178	<code>\MT@get@size</code> : additional magic to catch some errors . . . . .	108
add settings for Computer Modern Roman math symbols . . . . .	183	<code>\MT@get@size@</code> : hijack <code>\set@fontsize</code> instead of <code>\setfontsize</code> . . . . .	108
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement . . . . .	60	<code>\MT@loop</code> : fix: new macro, used instead of <code>\loop</code> . . . . .	49
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals) . . . . .	88	<code>\MT@maybe@do</code> : also check for alias font name . . . . .	60
<code>\MT@get@highlevel</code> : check whether defaults have changed . . . . .	106	<code>\MT@permute@@@@</code> : more sanity checks for <code>\SetProtrusion</code> and <code>\SetExpansion</code> . . . . .	122
		<code>\MT@setupfont</code> : also search for alias font file . . . . .	57
		fix: call <code>\@enc@update</code> if necessary . . . . .	57

### 2004/10/27 Version 1.3

General: fix: specifying <code>load</code> option does no longer require to give a name, too . . . . .	116	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german) . . . . .	35
Font aliases: declare <code>aer</code> , <code>zer</code> and <code>hfor</code> as aliases of <code>cmr</code> . . . . .	144	<code>\MT@load@list</code> : check whether list exists . . . . .	87

### 2004/11/12 Version 1.4

General: check for <code>pdfcprot</code> . . . . .	54	(OT1, T1, lmr) . . . . .	160
don't use scratch registers in global definitions . . . . .	91	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . .	130
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as the hook for <code>\MT@setupfont</code> . . . . .	99	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too . . . . .	113
use one instead of five counters . . . . .	50		
Protrusion: tweak quote characters for <code>cmr</code> variants			

### 2004/11/17 Version 1.4a

General: new option: <code>final</code> . . . . .	127	when reading files (reported by <i>Michael Hoppe</i> ) . . . . .	88
<code>\MT@cfg@catcodes</code> : fix: reset some more catcodes			

2004/11/26 **Version 1.4b**

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i> )	129	form abccz (reported by <i>Georg Verwey</i> )	88
optimisation: use less <code>\expandafers</code> and <code>\csnames</code>	44	<code>\MT@get@slot</code> : don't define <code>\MT@char</code> globally (save stack problem)	91
Protrusion: harmonise dashes in upshape and italic ( <code>cmr</code> , <code>pad</code> , <code>pp1</code> )	154	<code>\MT@ifdimen</code> : don't set <code>\MT@count</code> globally (save stack problem)	46
slanted like italics	163	<code>\MT@setup@PDF</code> : new message if <code>\pdfoutput</code> is changed	134
<code>\MT@checklist@family</code> : fix: don't try alias family name if encoding failed	61	<code>\MT@use@set</code> : don't use undeclared font sets	110
<code>\MT@get@basefamily</code> : fix: failed for font names of the			

2004/12/15 **Version 1.5**

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i> )	128	<code>\MT@get@highlevel</code> : don't test defaults if called after begin document	106
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i> )	126	<code>\MT@scale@factor</code> : warning for factors outside limits	66
Documentation: add 'Short history'	30	<code>\MT@scale@to@em</code> : don't use <code>\lcode</code> and <code>\rcode</code> for the calculation	64
Inheritance: remove <code>\ss</code> from T1 list, add <code>\DJ</code>	148	<code>\MT@set@ex@codes</code> : allow non-selected font expansion	70
Protrusion: settings for Bitstream Charter	155	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters	62
<code>\DeclareMicrotypeAlias</code> : remove spaces around arguments	111	<code>\MT@setup@expansion</code> : defaults: calculate step as $\min(\text{stretch}, \text{shrink})/5$	135
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'=</code> ' (compatibility with Turkish <code>babel</code> )	88	defaults: turn off expansion for DVI output	135
<code>\MT@fix@catcode</code> : reset catcode of <code>'^</code> ' (compatibility with <code>chemsym</code> )	35	disable automatic expansion for DVI output	136

2005/01/24 **Version 1.6**

General: defaults: turn off expansion for old pdfTeX versions	129	tune CMR math letters (OML encoding)	183
load a font if none is selected	56	<code>\MT@get@charwd</code> : use e-TeX's <code>\fontcharwd</code> , if available	64
new option: factor, by default 1000	128	<code>\MT@get@inh@list</code> : correct message if selected is false	91
restructure dtx file	143	<code>\MT@set@ex@codes</code> : introduce factor option	70
test whether <code>\pickup@font</code> has changed	102	<code>\MT@set@pr@codes</code> : introduce factor option	62
test whether numeric options receive a number	128	<code>\MT@setup@expansion</code> : disable automatic expansion for old pdfTeX versions	136
use e-TeX's <code>\ifcsize</code> and <code>\ifdefined</code> if defined	44	<code>\MT@use@set</code> : retain current set if new set is undeclared	110
Protrusion: add italic uppercase Greek letters	163	<code>\MT@vinfo</code> : new macro instead of <code>\ifMT@verbose</code>	36
improve settings for numbers (pointed out by <i>Peter Muthesius</i> )	156		

2005/02/02 **Version 1.6a**

Documentation: add table of fonts with tailored protrusion settings	21	reported by <i>Bernard Gaulle</i> )	91
<code>\MT@get@slot</code> : completely redone, hopefully more robust (compatible with <code>frenchpro</code> ; problem		<code>\MT@pdf@tex@no</code> : new macro	39
		<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older pdfTeX versions	70

2005/03/23 **Version 1.7**

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i> )	107	<code>\textbackslash</code> to T1 encoding	158
disallow automatic expansion if pdfTeX too old	119	<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i> )	111
fix: remove space after <code>autoexpand</code>	119	<code>\Microtype@Hook</code> : new command for font package authors	130
new value for verbose option: errors	128	<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	130
shorter command names	51	<code>\MT@begin@catcodes</code> : also use inside configuration commands	88
warning when running in draft mode	134	<code>\MT@cfg@catcodes</code> : reset catcode of <code>'=</code> ' (compatibility with <code>french*</code> packages)	88
Documentation: add hint about compatibility	26		
remove table of match order (now table 4 on page 89)	12		
Protrusion: fix: remove <code>\</code> from OT1, add			



\MT@DeclareMicrotypeAlias: may also be used inside configuration files	111	\MT@scale: new macro: use e-TeX's \numexpr if available	51
\MT@get@listname@: use \@tfor ( <i>Andreas Böhmann's</i> idea)	89	\MT@set@ex@codes: two versions of this macro	70
\MT@get@slot: remove backslash hack	91	\MT@split@name: don't define \MT@encoding &c.	
test for \chardefed commands	92	\globally	59
test whether \<encoding>\<...> is defined	91	\MT@test@ast: make it simpler	106
\MT@if@list@exists: don't define \MT@#1@c@name		\MT@try@order: always check for size, too (suggested by <i>Andreas Böhmann</i> )	89
\globally, here and elsewhere	90	fix: also check for //<series>/<shape>/ (reported by <i>Andreas Böhmann</i> )	89
\MT@if@dimen: comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i> )	46	\MT@warn@code@too@large: new macro: type out maximum protrusion factor	66
\MT@increment: use e-TeX's \numexpr if available	50	\MT@warn@err: new macro: for verbose=errors	36
\MT@is@composite: new macro: construct command for composite character; no uncontrolled expansion	97	\showhyphens: modify \showhyphens	137

2005/06/23 **Version 1.8**

General: \SetProtrusion: new key: unit	118	\MT@find@file: no longer wrap names in commands	87
if font substitution has occurred, set up the substitute font, not the selected one	99	\MT@fix@fontdimen@six: new macro: test whether \fontdimen 6 is defined	59
new option: config to load a different main configuration file	129	\MT@get@charwd: warning for missing (resp. zero-width) characters	64
new option: unit, by default character	128	\MT@get@listname@: made recursive	89
Documentation: add example for factor option	13	\MT@get@slot: fix: expand active characters	91
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i> )	15	test whether \<encoding>\<...> is defined made more robust	91
add hint about error messages	27	\MT@get@unit: new macro: get unit for codes	67
Font aliases: declare ppr and txr as aliases of ppl resp. ptm	145	\MT@in@rlist: made recursive	49
Font sets: add U encoding to allmath	143	\MT@is@active: new macro: translate inputenc-defined characters	95
Inheritance: remove \DJ from T1 list (it's the same as \DH)	148	\MT@is@letter: warning for non-ASCII characters	94
Protrusion: add LY1 characters for Times	163	\MT@ledmac@setup: character protrusion with ledmac	53
settings for AMS math fonts	187	\MT@map@clist@n: new macro: used instead of \@for	48
verified settings for slanted Computer Modern Roman	171	\MT@map@tlist@n: new macro: used instead of \@tfor	48
\add@accent: fix: disable micro-typographic setup inside \add@accent (reported by <i>Stephan Hennig</i> )	101	\MT@old@cmd: renamed commands from \..MicroType.. to \..Microtype..	36
\DeclareMicrotypeAlias: warning when overriding an alias font	111	\MT@pdf@tex@no: case 5: pdfTeX 1.30	40
\DeclareMicrotypeSetDefault: new command: set default font set	110	\MT@permute@@@@: add ranges to the beginning of the lists	122
\MT@cfg@catcodes: reset catcodes of the remaining ASCII characters	88	\MT@scale: fix: remove spaces in e-TeX variant (reported by <i>Mark Rossi</i> )	51
\MT@check@rlist: made recursive	123	\MT@setupfont@hook: restore % and # when hyperref is loaded	54
\MT@curr@list@name: new macro: current list type and name	98	restore csquotes's active characters	54
\MT@declare@sets: warning when redefining a set	105	restore percent character if Spanish babel is loaded	54
\MT@define@set@key@: use comma lists instead of token lists	105	\MT@split@codes: get character width once only	63
		\MT@use@set: fix: remove braces in first line	110
		\MT@xadd: simplified	47

2005/10/28 **Version 1.9**

General: \DeclareMicrotypeSet: new key: font	108	Documentation: add hint about verbatim environment	25
\SetProtrusion: value 'relative' renamed to 'character' for key unit	118	add remark about Type 1 fonts required for automatic font expansion	8
allow context-specific font setup	99	Font aliases: declare qpl and qtm (qfonts, TeX Gyre)	
compatibility with TeX Live hack (reported by <i>Herrbert Voß</i> )	39	as aliases of ppl resp. ptm	145
disable microtype setup inside hyperref's \pdfstringdef (reported by <i>Hàn Thế Thành</i> )	55	Font sets: add OT4 encoding to text sets	143
fix: use true as the default value	125	add T5 encoding to text sets	143
option unit: rename value relative to character	128	Inheritance: add list for OT4	149
		add list for T5 (requested by <i>Hàn Thế Thành</i> )	150

Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	158	<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists	67
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	154	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code>	95
settings for T5 encoded Computer Modern Roman	154	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code>	94
<code>\DisableLigatures</code> : new command: disable ligatures (requires pdfTeX 1.30)	112	<code>\MT@maybe@do</code> : redone	60
<code>\microtypecontext</code> : new command: change setup context in the document	103	<code>\MT@rem@from@clist</code> : new macro: remove an item from a comma list	48
<code>\MT@checklist@family</code> : fix: add two missing <code>\expandafters</code>	61	<code>\MT@scale@factor</code> : generalised	66
<code>\MT@detokenize@c</code> : fix the $\LaTeX$ version	45	<code>\MT@setup@expansion</code> : disable expansion if both step and shrink are zero	137
<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code>	44	warning if user requested zero step	135
		<code>\MT@toks</code> : use instead of <code>\toks@</code>	42
		<code>\SetProtrusion</code> : (et al.) new key: font	113

2005/12/05 **Version 1.9a**

General: ‘ <code>(file name)/(line number)</code> ’ as default list name	116	diately (requested by <i>Georg Verwey</i> )	105
new option: <code>defersetup</code> , by default true	127	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed	106
remove superfluous test whether <code>\pickup@font</code> has changed	102	<code>\MT@ifdefined@c@T</code> : new macros: true case only	45
Documentation: add explanation for error message in DVl mode	27	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available	45
add explanation for error message with non-Type 1 fonts	28	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available	47
Font aliases: declare <code>mbch</code> (mathdesign) as an alias of <code>bch</code>	146	<code>\MT@in@clist</code> : fix	48
Protrusion: fix: remove ‘_’ from OT1 encoding	159	<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)	65
settings for T5 encoded Charter	154	<code>\MT@is@feature</code> : new macro: check for pdfTeX feature	51
<code>\microtypesetup</code> : inside the preamble, accepts all package options	130	<code>\MT@map@clist@n</code> : following $\LaTeX$ 3	48
<code>\MT@check@font@cx</code> : optimise context-sensitive setup	102	<code>\MT@permute@</code> : don’t define permutations for unused encodings	122
<code>\MT@define@set@key@</code> : don’t expand variables immediately		<code>\MT@rem@from@clist</code> : fix	48
		<code>\MT@setup@</code> : defer setup until the end of the preamble	52

2006/01/20 **Version 1.9b**

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i> )	55	add samples of micro-typographic features	4
compatibility with the <code>extendedchar</code> option of the listings package	56	<code>\MT@features</code> : use throughout the package to adjust to beta-ness	51
Documentation: activate expansion in the distributed PDF	1	<code>\MT@ifdimen</code> : use <code>\pdfmatch</code> if available	46
		<code>\MT@warn@code@too@large</code> : fix calculation with present factor	66

2006/02/02 **Version 1.9c**

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i> )	22	<code>\MT@define@code@key@font</code> : fix: context was ignored	115
Protrusion: settings for URW Garamond	155	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in v1.9b)	115

2006/05/05 **Version 1.9d**

Font sets: <code>md*</code> instead of <code>m</code> series in basic sets	143	<code>\DeclareCharacterInheritance</code> : fix: empty context	119
add QX encoding to text sets	143	<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available	45
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> )	149	<code>\MT@get@ex@opt</code> : fix: evaluate preset	71
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> )	161	<code>\MT@get@font@dimen</code> : warning for zero fontdimen	65
settings for Euro symbols (Adobe, ITC, marvosym)	194	<code>\MT@get@opt</code> : optimise: don’t reset when preset option is set	67
tweak AMS settings	187	set list name before presetting	67

<code>\MT@is@active</code> : support for Unicode (inputenc/utf8) . . .	95	mandatory argument . . . . .	113
<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>tex4ht</code> is loaded (reported by <i>Peter Dyballa</i> ) . . . . .	54	(et al.) split keys of optional and mandatory argument . . . . .	113
<code>\SetProtrusion</code> : (et al.) optimise: unify keys for			

2006/07/28 **Version 1.9e**

General: fix: default value for <code>activate</code> : <code>true</code> . . .	126	settings for Euler Roman font . . . . .	190
Documentation: add hint about unknown encodings include LPPL . . . . .	26 249	<code>\DeclareCharacterInheritance</code> : new key ‘inputenc’ to set the input encoding . . . . .	119
Font aliases: declare <code>zeur</code> and <code>zeus</code> (eulervm) as aliases of <code>eur</code> resp. <code>eus</code> (euler) . . . . .	146	<code>\MT@rem@from@clist</code> : model after <code>\@removeelement</code> . . . . .	48
Inheritance: adapt to <code>marvosym</code> ’s changed encoding . . . . .	151	<code>\MT@setup@</code> : empty <code>\MT@setup@</code> after use (compatibility with the <code>combine</code> class) . . . . .	52
Protrusion: complete settings for Euler Fraktur and Script fonts . . . . .	194	<code>\pickup@font</code> : no tracing with <code>trace</code> package . . . . .	101
fix: forgotten comma in <code>mt-mvs.cfg</code> ; adapt to <code>marvosym</code> ’s changed encoding . . . . .	194	<code>\SetExpansion</code> : new key: <code>inputenc</code> . . . . .	113
		<code>\SetProtrusion</code> : (et al.) new key: <code>inputenc</code> . . . . .	113

2006/09/09 **Version 1.9f**

Protrusion: fix: <code>euler-vm</code> did not load euler settings . . .	191	<code>\MT@reset@context</code> : only reset context if it has actually been changed . . . . .	103
<code>\MT@curr@list@name</code> : fix: <code>\MessageBreak</code> must not be expanded . . . . .	98	<code>\MT@set@inh@list</code> : fix: forgotten comma in the features list . . . . .	120
<code>\MT@gdef@n</code> : new macros: global variants . . . . .	44	<code>\MT@set@named@keys</code> : new macro: set name first, simplify parsing of optional argument . . . . .	114
<code>\MT@get@inh@list</code> : fix: input encoding must be set after the inheritance list has been parsed . . . . .	91	<code>\SetProtrusion</code> : (et al.) set catcodes before parsing optional argument . . . . .	113
<code>\MT@glet</code> : new macro . . . . .	44		

2007/01/14 **Version 2.0**

General: compatibility with listings: set catcode of backslash to zero (reported by <i>Steven Bath</i> ) . . .	55	new: <code>smallcaps</code> . . . . .	143
compatibility with <code>soul</code> : register <code>\textls</code> and <code>\lststyle</code> . . . . .	56	<code>\DeclareMicrotypeBabelHook</code> : new command: interaction with <code>babel</code> . . . . .	112
new option: <code>babel</code> , by default <code>false</code> (language-dependent setup suggested by <i>Ulrich Dirr</i> ) . . .	126	<code>\lststyle</code> : fix: font switches don’t pose a problem anymore . . . . .	79
new option: <code>letterspace</code> , by default <code>100</code> . . . . .	128	fix: <code>letterspacing</code> commands may be nested . . . . .	79
new package <code>letterspace</code> : a stripped-down version, containing the <code>letterspacing</code> commands only . . . . .	1	new command: <code>letterspacing</code> . . . . .	79
option ‘ <code>babel</code> ’: fix: switch off French <code>babel</code> ’s short-hands properly (reported by <i>Daniel Flipo</i> ) . . .	141	totally redone, using the new <code>\letterspacefont</code> . . . . .	79
option ‘ <code>babel</code> ’: switch off Turkish <code>babel</code> ’s short-hands . . . . .	142	<code>\MT@declare@sets</code> : fix: empty size list when redefining set . . . . .	105
option ‘ <code>unit</code> ’, <code>\SetProtrusion</code> : deprecate value ‘ <code>relative</code> ’ completely . . . . .	118	<code>\MT@is@symbol</code> : made even more robust . . . . .	96
Documentation: add hint about how to increase <code>font_max</code> and <code>font_mem_size</code> . . . . .	28	<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings) . . . . .	68
add hint about warning when tracking and expansion is applied to a font . . . . .	28	<code>\MT@pdftex@no</code> : case 6: <code>pdfTeX 1.40</code> . . . . .	40
add remark about ‘ <code>draft</code> ’ option disabling microtype (noted by <i>Michalis Miatidis</i> ) . . . . .	9	<code>\MT@setup@noligatures</code> : maybe disable <code>\MT@noligatures</code> after the preamble . . . . .	140
qualify hint about web documents with regard to older <code>pdfTeX</code> versions . . . . .	25	<code>\MT@split@name</code> : adjust to possible <code>letterspacing</code> . . . . .	59
qualify hints about expansion error messages with regard to older <code>pdfTeX</code> versions . . . . .	27	<code>\SetExtraKerning</code> : new command: additional kerning . . . . .	114
Font sets: new: <code>footnotesize</code> and <code>scriptsize</code> . . .	143	<code>\SetExtraSpacing</code> : new command: adjustment of interword spacing . . . . .	114
		<code>\SetTracking</code> : new command: tracking . . . . .	113
		<code>\textls</code> : new command: <code>letterspacing</code> . . . . .	83
		starred version: remove spaces around text . . . . .	83
		<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations . . . . .	37

2007/01/21 **Version 2.1**

General: compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i> )	56	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	79
fix: letterspace package forgot to load <code>keyval</code>	42	<code>\MT@orig@pickupfont</code> : compatibility with CJK: also check for its definition	100
<code>\slig</code> : new command: protect ligatures in letter-spaced text	79	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	83

2007/07/14 **Version 2.2**

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i> )	52	<code>\MT@is@composite</code> : more robust: expand exactly once	97
new option: <code>copyfonts</code>	127	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	96
simplify key declarations	116	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	75
use catcode trickery for e-TeX test	38	<code>\MT@lua</code> : (basic) support for LuaTeX	41
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@pdftex@no</code> : case 7: pdfTeX 1.40.4	40
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i> )	25	<code>\MT@preset@aux@space</code> : generalised	68
add overview	5	<code>\MT@set@all@pr</code> : (et al.) allow empty values	63
logo transparency and amusement	1	<code>\MT@set@inputenc@</code> : only load inputenc files if necessary	68
Font aliases: declare <code>chr</code> ( <code>chmath</code> ) as an alias of <code>bch</code> (reported by <i>Geoff Vallis</i> )	146	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	76
declare <code>fp9x</code> , <code>fp9j</code> (FPL Neu) as aliases of <code>ppl[xj]</code>	145	possibility to customise interword spacing	76
Font sets: default set for tracking: <code>smallcaps</code>	144	<code>\MT@setup@expansion</code> : warning if stretch or shrink aren't multiples of step	137
Inheritance: remove <code>'-'</code> → <code>'127'</code>	148	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	57
Protrusion: settings for Bitstream Letter Gothic	155	only add features that are available with the respective pdfTeX	57
Spacing: add sample	195	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	54
Tracking: add ligatures that are to be disabled	152	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	63
<code>\DeclareMicrotypeVariants</code> : new command	111	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	75
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	112	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	75
<code>\slig</code> : always defined	79	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	140
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	62	<code>\MT@with@babel@and@T</code> : also inspect class options	52
<code>\MT@copy@font</code> : optionally work on copies of fonts	58	<code>\pickup@font</code> : letterspace: setup inside group	101
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	88	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	113
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	65	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i> )	113
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	79	third argument may be empty	113
<code>\MT@ifdimen</code> : employ LuaTeX features if available	46	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	103
<code>\MT@ifint</code> : employ LuaTeX features if available	45		
<code>\MT@ifstreq</code> : employ LuaTeX features if available	47		
fix: $\epsilon$ -TeX version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i> )	47		

2007/12/23 **Version 2.3**

General: disable <code>\microtypecontext</code> in <code>hyperref's</code> <code>\pdfstringdef</code>	55	<code>\microtypecontext</code> : made robust (reported by <i>Stephan Hennig</i> )	103
fix: really switch off Turkish shorthands	142	<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code>	88
new value for verbose option: <code>silent</code> (suggested by <i>Karl Berry</i> )	128	<code>\MT@define@set@key@font</code> : font: single asterisk means normal font	108
turned some warnings into errors	127	<code>\MT@in@clist</code> : don't use <code>\x</code> (reported by <i>Peter Meier</i> )	48
Documentation: add kerning sample	18	<code>\MT@is@active</code> : support for extended Unicode (inputenc/utf8x resp. ucs) – experimental	95
add letterspacing illustration	17	<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in v2.2)	85
<code>\do@subst@correction</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i> )	101	<code>\MT@plain</code> : letterspace: support for <code>eplain/miniltx</code>	38
<code>\slig</code> : redone: extract outer kerns from current letterspacing amount	79	<code>\MT@set@curr@eos</code> : adjusting spaces made more reliable	77

<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> )	76	<code>\MT@SetTracking</code> : sanity check for value	114
possibility to customise outer kerning (suggested by <i>Stephan Hennig</i> )	76	<code>\MT@setup@tracking</code> : enable protrusion when tracking is enabled	138
		<code>\MT@tr@outer@l</code> : only change pre outer space if it contains shrink	81

2008/02/29 **Version 2.3a**

General: fix test for <code>soul</code> under plain $\TeX$	56	too old for extensions	139
Documentation: add hint about <code>babel</code> having to be loaded first	27	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the <code>letterspace</code> package	35
add table of available and enabled features	7	<code>\MT@getkey</code> : fix: <code>key=val</code> in class options list	133
mention <code>soulutf8</code>	30	<code>\MT@set@codes</code> : generalised	63
Protrusion: adjust LMR quotation marks again	160	<code>\MT@setupfont@hook</code> : restore percent character if Mexican <code>babel</code> is loaded	54
<code>\MT@error@doesnt@work</code> : error messages if <code>pdfTeX</code> is			

2008/06/04 **Version 2.3b**

<code>\MT@exp@gcs</code> : new macro: reduce save stack size	44	also check for its definition	100
<code>\MT@font@copy</code> : enable font copies also with protrusion contexts (reported by <i>Nathan Rosenblum</i> )	58	<code>\MT@requires@latex</code> : new macro	38
<code>\MT@get@size@</code> : grouping	108	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i> )	76
<code>\MT@noligatures@</code> : fix: warning messages for unknown slots	86	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode	81
<code>\MT@orig@pickupfont</code> : compatibility with <code>CJKutf8</code> :		make <code>\spaceskip-aware</code> ( <code>ragged2e</code> )	81
		<code>\MT@tr@outer@r@</code> : additional test for horizontal mode	82

2008/11/11 **Version 2.3c**

General: $\text{Lua}\TeX$ supported by default	40	coding (reported by <i>Vasile Gaburici</i> )	150
Documentation: add hint about spacing being experimental	26	<code>\MT@detokenize@c</code> : fix: remove last space only (reported by <i>Ulrich Durr</i> )	45
add hint about partial incompatibility with <code>CJK</code>	27	<code>\MT@tr@outer@r@</code> : additional test for horizontal mode (reported by <i>Sveinung Heggen</i> )	82
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> )	56	(reported by <i>Ulrich Durr</i> )	80
move setup to the very end (for <i>Colin Rourke</i> )	142	<code>\MT@setup@expansion</code> : default step: 1 for <code>pdfTeX</code> versions $\geq 1.40$	135
<code>\ifMT@inannot</code> : use <code>pdfTeXcmds</code> for debugging	37	<code>\MT@tr@outer@r@</code> : don't use <code>\x</code> (reported by <i>Ulrich Durr</i> )	82
<code>\lststyle</code> : disable for $\text{Lua}\TeX$	79	fix: don't adjust in math mode (reported by <i>Christoph Bier</i> )	82
make invalid in math mode	79	fix: don't adjust inside discretionary (reported by <i>Maverick Woo</i> )	82
<code>\microtypesetup</code> : select font after setup	130	<code>\MT@tr@set@okern</code> : allow empty value for outer kerning	84
<code>\MT@check@active@set</code> : warning for missing default sets	130	<code>\textls</code> : make math mode aware	83
<code>\MT@lua</code> : update for $\text{Lua}\TeX$ 0.36	41		
<code>\MT@set@tr@codes</code> : allow zero tracking	75		
<code>\MT@set@tr@zero</code> : fix: allow switching off tracking			

2009/11/09 **Version 2.3e**

Documentation: suggest to patch <code>\@verbatim</code> instead of <code>\verbatim</code>	26	<code>\MT@fix@fontdimen@six</code> : fix: gobbling settings with tracking failed (reported by <i>Leo</i> )	59
Expansion: settings for T2A encoding (contributed by <i>Karl Karlsson</i> )	153	<code>\MT@setup@</code> : make space-unaware (requested by <i>Marcin Borkowski</i> )	52
Font sets: <code>sc*</code> instead of <code>sc</code> in <code>smallcaps</code> set	143	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (first reported by <i>Christian Stark</i> )	54
add T2A encoding	143	<code>\MT@tr@outer@r@</code> : fix: set current kerning and spacing again (found by <i>Lars Rönnbäck</i> )	82
Protrusion: settings for T2A encoding (contributed by <i>Karl Karlsson</i> )	160		
Spacing: settings for T2A encoding (contributed by			



2010/01/10 **Version 2.4**

- General: new file `microtype.lua` containing the lua functions (contributed by *Élie Roux*) ..... 43
- Protrusion: settings for T2A encoded Minion (contributed by *Karl Karlsson*) ..... 160

2013/03/13 **Version 2.5**

- General: allow contexts for LuaTeX ..... 117
- disable ‘DVIOutput’ option for XeTeX ..... 126
- fix: check whether ‘*{file}/<line>*’ list name already exists (reported by *Till A. Heilmann*) ..... 116
- letterspacing with LuaTeX 0.62 ..... 74
- new files: `microtype-pdftex.def`, `microtype-xetex.def`, `microtype-luatex.def`, containing engine-specific definitions ..... 38
- protrusion with XeTeX ..... 40
- restore `\space` inside listings (reported by *Rolf Dieterich*) ..... 55
- Documentation: add hint about LuaTeX compatibility ..... 27
- add hint about spacing and `ragged2e` ..... 26
- add hint about dtx source code ..... 28
- include `microtype-logo.dtx` and `microtype-lssample.dtx` ..... 221
- Font aliases: declare `\msy` and `\lmm` as aliases of `\cmsy` resp. `\cmm` (reported by *Jonas Hogstrom*) ..... 144
- declare `\zgm` etc. (`\garamondx`) as aliases of `\ugm` ..... 146
- declare Latin Modern Roman (OpenType version) as alias of `\lrm` when `fontspec` is loaded ..... 144
- declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) ..... 145
- Font sets: add EU1 and EU2 encodings ..... 143
- Inheritance: add rudimentary list for EU1 and EU2 ..... 151
- Protrusion: add default lists for EU1 and EU2 ..... 159
- improvements to Computer Modern Roman italics (contributed by *Hendrik Vogt*) ..... 163
- Tracking: add EU2 encoding to default list ..... 152
- `\DeclareCharacterInheritance`: allow more than one encoding ..... 119
- `\DeclareMicrotypeAlias`: ignore spaces ..... 111
- `\ifMT@nofamily`: info if settings are not family-specific (suggested by *Hàn Thê Thành*) ..... 62
- `\LoadMicrotypeFile`: remove all spaces in font name ..... 111
- `\lstyle`: fix: ensure to set up math fonts (reported by *RazorXsr*) ..... 79
- `\MT@define@code@key@family`: compatibility with `fontspec`: remove its internal counter (reported by *Till A. Heilmann*) ..... 115
- `\MT@define@code@key@font`: scrub `fontspec` feature count (found by *Meho R*) ..... 115
- `\MT@do@font`: adapt for LuaTeX ..... 50
- adapt for XeTeX ..... 50
- `\MT@get@slot@`: adapt for LuaTeX (requested by *Georg Duffner*) ..... 92
- adapt for XeTeX ..... 92
- `\MT@if@outer@next`: fix: conflict with `amsmath` (reported by *Scott Pakin*) ..... 82
- `\MT@info@missing@char`: fix error message for XeTeX (reported by *Juan Acevedo*) ..... 65
- `\MT@is@charx`: compatibility with unicode ..... 96
- `\MT@ledmac@setup`: fix to work with XeTeX (reported by *Maieul Rouquette*) ..... 53
- `\MT@ls@set@ls`: allow formulas in optional argument to `\textls` (fix by *Heiko Oberdiek*) ..... 84
- `\MT@microtypecontext`: fix: ensure to set up math fonts (reported by *RazorXsr*) ..... 103
- `\MT@register@subst@font`: only register substituted font if it isn’t registered already (reported by *George Gratzner* and *Josep Maria Font*) ..... 102
- `\MT@register@subst@font@cx`: only register if it isn’t registered already ..... 102
- `\MT@scrubfeatures`: compatibility with `fontspec`: remove its internal counter ..... 60
- `\MT@set@all@pr`: fix: remove space (found by *Meho R*) ..... 63
- `\MT@set@pr@codes`: make info about generic settings encoding-specific (reported by *Sebastian Schubert*) ..... 62
- `\MT@setup@spacing`: warning with `ragged2e` (reported by *Steffen Hoffmann*) ..... 138
- `\MT@setupfont`: select font with `fontspec` (found by *Georg Duffner*) ..... 56
- `\MT@setupfont@hook`: restore `\%` and `\#` when `mathastext` is loaded (found by *Seamus Bradley*) ..... 54

2013/05/23 **Version 2.5a**

- General: use `luatexbase` instead of `luatextra` (contributed by *Élie Roux*) ..... 43
- Documentation: add notes on typesetting the documentation ..... 28
- include OpenType configuration files ..... 201
- `\MT@afteraftergroup`: fix: get outer kerning and spacing of nested letterspacing right ..... 77
- `\MT@get@slot@`: adapt to `luaotfload v2.2` (contributed by *Élie Roux*) ..... 93
- `\MT@led@unhbox@line`: simplified ..... 53
- `\MT@ledmac@setup`: support for `eledmac` ..... 53
- `\MT@ls@outer@k`: add marker for tightly nested letterspacing ..... 85
- `\MT@set@tr@codes`: fix: load font for `fontspec` ..... 76
- `\MT@xspace`: fix outer spacing problem with `xspace` (reported by *Dave*) ..... 83

2016/05/01 **Version 2.6**

- General: load `luaotfload` with LuaTeX ..... 43
- redefine `\MT@setupfont@hook` globally for problem with `tikzposter` (reported by *Sam Mason*) ..... 55
- Documentation: add hint about partial incompatibility with `xeCJK` and `luatexja` ..... 27
- missing characters printed with `Charis SIL` ..... 201

suggest to use etoolbox to patch <code>\verbatim</code> . . . .	26	<code>\MT@is@xchar</code> : update for fontspec's TU encoding .	96
Font sets: add TU encoding (notified by <i>Will Robertson</i> ) . . . . .	143	<code>\MT@ledmac@setup</code> : support for <code>reledmac</code> . . . . .	53
add <code>si</code> and <code>scit</code> to <code>smallcaps</code> set (reported by <i>uli</i> )	143	<code>\MT@luatex@no</code> : update for LuaTeX 0.85 (renamed primitives) . . . . .	41
new: <code>allmath-nott</code> and <code>alltext-nott</code> (suggested by <i>Karl Berry</i> ) . . . . .	143	<code>\MT@noligatures@</code> : use <code>luaotfload</code> function to keep/inhibit ligatures . . . . .	86
Inheritance: add TU encoding . . . . .	151	<code>\MT@orig@pickupfont</code> : (in)compatibility with <code>luatexja</code> : disable unknown slots warnings (reported by <i>Max</i> ) . . . . .	100
Protrusion: add TU encoding to lists . . . . .	159	(in)compatibility with <code>xeCJK</code> : disable unknown slots warnings (reported by <i>HcN</i> ) . . . . .	100
Tracking: add TU encoding to default list . . . . .	152	compatibility with <code>xeCJK</code> : pretend that CJK wasn't loaded . . . . .	100
<code>\DeclareMicrotypeSet</code> : ignore spaces . . . . .	105	<code>\MT@set@tr@codes</code> : use <code>luaotfload</code> 's <code>kernfactor</code> feature if available . . . . .	76
<code>\DeclareMicrotypeSetDefault</code> : ignore spaces . . .	110	<code>\MT@xspace</code> : fix outer spacing problem with (not only) algorithm (reported by <i>Henning</i> and <i>Ronnie Marks</i> ) . . . . .	83
<code>\DeclareMicrotypeVariants</code> : ignore spaces . . . .	111	<code>\UseMicrotypeSet</code> : ignore spaces . . . . .	109
<code>\lststyle</code> : fix: ensure to set up math fonts (reported by <i>kleenstar</i> ) . . . . .	79		
<code>\microtypecontext</code> : allow activate shortcut (reported by <i>Karl Berry</i> ) . . . . .	103		
<code>\MT@declare@sets</code> : fix: undefine lists for redefining	105		
<code>\MT@do@font</code> : speed up for LuaTeX . . . . .	50		
<code>\MT@engine</code> : fix test with LuaTeX 0.85 . . . . .	39		
<code>\MT@get@slot@</code> : fix: could fail with XeTeX (reported by <i>Christopher Schramm</i> ) . . . . .	92		

2016/05/14 **Version 2.6a**

General: fixes for <code>letterspace</code> package with LuaTeX	49	<code>\Voß</code> . . . . .	50
<code>\MT@do@font</code> : fix <code>lua</code> function (reported by <i>Herbert</i>		<code>\MT@ls@fontspec@font</code> : fix for value of $\pm 1000$ . . .	78

2017/07/07 **Version 2.7**

General: drop <code>luatexbase</code> with recent L <sup>A</sup> T <sub>E</sub> X . . . .	43	<code>\MT@check@range@</code> : don't warn for override if conflicting list is loaded . . . . .	124
warning with <code>minimal</code> class . . . . .	52	<code>\MT@is@composite</code> : compatibility with L <sup>A</sup> T <sub>E</sub> X 2017/01/01 ( <code>\DeclareUnicodeComposite</code> ) (reported by <i>Ulrike Fischer</i> and <i>jcr</i> ) . . . . .	97
Documentation: mention that additional kerning does not work in math mode (discovered by <i>Daniel</i> )	18	<code>\MT@ls@fontspec@font</code> : fix for 'file:{font}' spec (reported by <i>Reinhard Kotucha</i> ) . . . . .	78
Font aliases: declare aliases for <code>newpx</code> . . . . .	145	<code>\MT@permute@@@@</code> : don't warn for override if conflicting list is loaded . . . . .	123
declare aliases for <code>newtx</code> . . . . .	145	<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older LuaTeX versions . . . . .	70
declare aliases for <code>tempora</code> . . . . .	145	<code>\MT@setup@expansion</code> : don't disable automatic expansion for DVI output with LuaTeX . . . . .	136
declare aliases for <code>XCharter</code> . . . . .	146	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (again) . .	54
declare Latin Modern Roman as alias of <code>lmr</code> with new L <sup>A</sup> T <sub>E</sub> X format (reported by <i>Ulrike Fischer</i> )	144	<code>\MT@warn@tracking@DVI</code> : don't warn for letterspacing in DVI mode with LuaTeX . . . . .	140
Protrusion: automatically choose correct names for Charis SIL small caps (reported by <i>'ltcomdata'</i> )	214		
<code>\lststyle</code> : fix: prevent infinite loop with <code>psnfss</code> and <code>exscale</code> packages (reported by <i>user11126</i> , solution by <i>Ulrike Fischer</i> ) . . . . .	79		

2018/01/14 **Version 2.7a**

General: disallow non-automatic expansion with LuaTeX . . . . .	119	<code>\MT@get@highlevel</code> : test whether <code>\...default</code> is defined . . . . .	106
<code>\MT@auto</code> : remove 'autoexpand' for LuaTeX 1.0.6 (reported by <i>Ulrike Fischer</i> ) . . . . .	136	<code>\MT@get@slot</code> : expand active characters earlier . . .	91
with LuaTeX, non-automatic font expansion is no longer possible (as confirmed by <i>Hans Hagen</i> )	136	<code>\MT@info@nottracking@</code> : defer 'No tracking' message	60
		<code>\MT@is@active</code> : compatibility with <code>newunicodechar</code> (reported by <i>Nils Anders Danielsson</i> ) . . . . .	95

2019/02/28 **Version 2.7b**

General: update <code>lua</code> function <code>microtype.info</code> after changes in <code>luaotfload</code> (reported by <i>Moritz Wemheuer</i> and <i>Ulrike Fischer</i> ) . . . . .	43	(reported by <i>Franz Wexler</i> ) . . . . .	149
Documentation: update hint about non-7-bit characters (notified by <i>Frank Mittelbach</i> ) . . . . .	26	<code>\MT@info@missing@char</code> : fix message for glyphs specified as names in XeTeX (reported by <i>Paolo Ney</i> )	65
Inheritance: add <code>textquotedblleft</code> ligature to OT4		<code>\MT@setupfont</code> : always select current font with XeTeX and LuaTeX (reported by <i>Paolo Ney</i> , solution by <i>Ulrike Fischer</i> ) . . . . .	56

2019/10/10 **Version 2.7c**

- General: turn warning into info when overwriting the  
keepligature function (reported by *Andy N*) .. 86
- `\MT@is@active`: compatibility with L<sup>A</sup>T<sub>E</sub>X 2019/10/01 95
- `\MT@is@symbol`: take care of `\remove@tlig` ..... 96
- `\showhyphens`: compatibility with L<sup>A</sup>T<sub>E</sub>X 2019/10/01  
(reported by *Phelype Oleinik* and *Falk Hanisch*) 137

2019/11/18 **Version 2.7d**

- `\MT@copy@font@:` in LuaT<sub>E</sub>X, don't use the `\copyfont`  
primitive, but load the font anew (reported by  
*Paolo Polesana* and *Oliver Kopp*) ..... 58
- `\MT@register@subst@font:` remove substitute font  
from list (reported by *Markus Kohm*) ..... 102
- `\MT@register@subst@font@cx:` remove substitute font  
from lists ..... 102

2020/12/07 **Version 2.8**

- General: letterspace works with e-T<sub>E</sub>X only ..... 38
- compatibility with soul: patch for font change  
(reported by *Md Ayquassar*) ..... 56
- fix for luatexbase ..... 43
- Documentation: declare DVIoutput option deprec-  
ated ..... 9
- squash fake news about automatic font expansion  
with dvilualatex ..... 8
- Font aliases: declare aliases for step and domitian  
(notified by *Daniel Benjamin Miller*) ..... 145
- declare aliases for stix and stix2 fonts ..... 145
- declare New Computer Modern as an alias of Latin  
Modern Roman ..... 145
- Font sets: default set for expansion: alltext-nott ..... 144
- default set for spacing: alltext-nott ..... 144
- `\lsstyle`: fix: enforce math setup, again ..... 79
- `\microtypecontext`: fix activate shortcut ..... 103
- ignore spaces ..... 103
- `\MT@do@font:` fix for X<sub>Y</sub>T<sub>E</sub>X ..... 50
- simplify lua function ..... 50
- `\MT@fix@fontdimen@six:` try to fix zero `\fontdimen 6` ..... 59
- `\MT@if@luaotf@font:` use lua function ..... 49
- `\MT@ifstreq:` use X<sub>Y</sub>T<sub>E</sub>X's `\stringcmp` ..... 47
- `\MT@setup@expansion:` warning when expanding in  
DVI mode with LuaT<sub>E</sub>X (reported by *Daniel Ben-  
jamin Miller*) ..... 135
- `\MT@tr@set@space@:` simplified ..... 80
- `\MT@tr@unit@:` fix: allow unit regardless whether  
letterspacing is set ..... 78
- `\textmicrotypecontext`: ignore spaces ..... 103

2020/02/22 **Version 2.8a**

- General: rename mt-pad.cfg to mt-EBGaramond.cfg  
(requested by *Karl Berry*) ..... 143
- rename mt-PalatinoLinotype.cfg to  
mt-Palatino.cfg (requested by *Karl Berry*) .. 201
- Documentation: reorganise table 1 for clarification  
(suggested by *Daniel Benjamin Miller*) ..... 7
- Font aliases: add -LF, -TLF, -OsF and -T0sF as vari-  
ants ..... 144
- declare mmodern fonts as aliases of Latin Modern  
(reported by *Daniel Benjamin Miller*) ..... 144
- reference New Computer Modern also by file name  
(reported by *Canageek*) ..... 145
- Inheritance: specify 'ff' ligature as Unicode instead of  
glyph name ..... 201
- Protrusion: hide euroitc settings (requested by *Karl  
Berry*) ..... 194
- Spacing: add dummy settings (reported by  
*DORpapt*) ..... 198
- `\DeclareCharacterInheritance`: skip settings for  
tracking ..... 119
- `\MT@get@size@@:` compatibility with svjour3 (repor-  
ted by *Ekkehart Schlicht* (no relation!)) ..... 108
- `\MT@gl@et:` use LuaT<sub>E</sub>X's `\gl@et`, if available ..... 44

2021/02/25 **Version 2.8b**

- Inheritance: dummy settings for the Font Awesome  
font (mt-FontAwesome.cfg) ..... 205
- settings for the Lato font (mt-Lato.cfg) (reported  
by *dsedivec*) ..... 205

**D Index**

Numbers in upright shape refer to the page where the corresponding entry is described (bold face) resp. occurs.  
Numbers in italics refer to the code line where the corresponding entry is defined (underlined) resp. used.

<b>Options</b>	DVIoutput ..... 9	config ..... 9
	activate ..... 6	draft ..... 9
	auto ..... 7	expansion ..... 6
	babel ..... 9	factor ..... 7



- final ..... 9  
 kerning ..... 6  
 letterspace ..... 8  
 protrusion ..... 6  
 selected ..... 8  
 shrink ..... 8
- Commands**
- \DeclareCharacterInheritance ..... 20  
 \DeclareMicrotypeAlias ..... 21  
 \DeclareMicrotypeBabelHook ..... 23  
 \DeclareMicrotypeSet\* ..... 10  
 \DeclareMicrotypeSet ..... 10  
 \DeclareMicrotypeSetDefault ..... 12  
 \DeclareMicrotypeVariants\* ..... 20  
 \DeclareMicrotypeVariants ..... 20  
 \DisableLigatures ..... 24  
 \LoadMicrotypeFile ..... 22  
 \SetExpansion ..... 14  
 \SetExtraKerning ..... 18
- A**
- aoposter (package) ..... 108  
 activate (option) ..... 6, 126, 235  
 \add@accent ..... 2945  
 \adjustspacing ..... 196, 4349  
 ae (package) ..... 21, 144
- B**
- babel (option) ..... 9, 23, 27, 32, 126, 235  
 babel (package) ..... 2, 3, 5, 17, 23, 24, 27,
- C**
- chemsym (package) ..... 232  
 chmath (package) ..... 21, 146, 236  
 CJK (package) ..... 27, 56, 100, 236, 237, 239  
 CJKutf8 (package) ..... 100, 237  
 cm-super (package) ..... 8  
 color (package) ..... 134  
 combine (package) ..... 142, 235
- D**
- \DeclareCharacterInheritance ..... 20, 41, 3786  
 \DeclareMicrotypeAlias ..... 21, 43, 71, 3378  
 \DeclareMicrotypeBabelHook ..... 23, 45, 64, 3444  
 \DeclareMicrotypeSet ..... 10, 32, 57, 59, 72, 3077  
 \DeclareMicrotypeSet\* ..... 10, 3077  
 \DeclareMicrotypeSetDefault ..... 12, 34, 61, 3333  
 \DeclareMicrotypeVariants ..... 20, 42, 58, 63, 3363  
 \DeclareMicrotypeVariants\* ..... 20  
 defersetup (option) ..... 54, 126, 127, 130, 234
- E**
- eco (package) ..... 21, 144  
 \efcode ..... 1555, 1581, 1582, 1623, 1625  
 eledmac (package) ..... 53, 238  
 eplain (package) ..... 24, 31, 34, 38, 52, 133, 236  
 e-TeX (engine) ..... 33, 38, 40,  
 45, 50, 51, 64, 82, 108, 221, 232–234, 236, 240  
 etoolbox (package) ..... 26, 239  
 eucal (package) ..... 192  
 eufrak (package) ..... 194
- F**
- \f@family ..... 2896, 2897  
 \f@size ..... 1929, 2889, 2891, 2893, 2900, 2943  
 factor (option) ..... 7, 13, 33, 128, 232  
 fancyvrb (package) ..... 26  
 final (option) ..... 9, 33, 126, 127, 231  
 fix-cm (package) ..... 59  
 \font ..... 970, 1046, 1831, 1979  
 \font@name ..... 111, 1018, 1020, 1022, 1024,  
 1025, 1035, 1036, 1821, 1829, 1833, 1837,  
 1875, 1881, 1886, 1889, 1964, 1979, 1991,  
 1993, 1998, 2000, 2003, 2006, 2009, 2044,  
 2095, 2232, 2234, 2882, 2890, 2899, 2904,  
 2924, 2926, 2928, 2944, 2958, 2960, 2980, 2985
- spacing ..... 6  
 step ..... 8  
 stretch ..... 8  
 tracking ..... 6  
 unit ..... 7  
 verbose ..... 9
- \SetExtraSpacing ..... 19  
 \SetProtrusion ..... 13  
 \SetTracking ..... 15  
 \UseMicrotypeSet ..... 12  
 \lslig ..... 24  
 \lststyle ..... 23  
 \microtypecontext ..... 22  
 \microtypesetup ..... 9  
 \textls\* ..... 24  
 \textls ..... 23  
 \textmicrotypecontext ..... 22
- algorithm (package) ..... 239  
 amsmath (package) ..... 238  
 amssymb (package) ..... 187  
 article (package) ..... 22  
 auto (option) ..... 7, 8, 126  
 54, 88, 112, 141, 142, 146, 232, 233, 235–237
- config (option) ..... 9, 20, 32, 129, 233  
 contour (package) ..... 134  
 copyfonts (option) ..... 58, 126, 127, 236  
 crop (package) ..... 134  
 csquotes (package) ..... 29, 54, 158, 233  
 \curr@fontshape ..... 2889, 2891, 2893, 2900, 2943  
 \currentiflevel ..... 3233, 3236
- \define@newfont ..... 2882, 2890, 2894, 2904  
 \DisableLigatures ..... 24, 40, 62, 3417  
 \do@subst@correction ..... 2942  
 docstrip (package) ..... 34  
 domitian (package) ..... 21, 145, 240  
 draft (option) ..... 9, 126, 127, 235  
 dsfont (package) ..... 59  
 DVIPoutput (option) ..... 9, 126, 238, 240
- euler (package) ..... 98, 190, 235  
 eulervm (package) ..... 21, 146, 191, 235  
 euroitc (package) ..... 194, 240  
 europus (package) ..... 194  
 eurosans (package) ..... 194  
 \expandglyphsinfont ..... 1537  
 expansion (option) ..... 6, 11, 125, 126, 135  
 exscale (package) ..... 239
- \fontcharwd ..... 1283, 1287, 1293  
 \fontdimen ..... 1050, 1051,  
 1055, 1056, 1059, 1318, 1323, 1423, 1426,  
 1850, 1851, 1881, 1991, 2037, 2038, 2045,  
 2046, 2053, 2055, 2061, 2076, 2095, 2245, 2248  
 fontinst (package) ..... 198, 221, 223  
 fontinstallationguide (package) ..... 177  
 \fontname ..... 1829, 1833  
 fontspec (package) ..... 6, 25, 27, 30, 31,  
 43, 59, 60, 78, 96, 106, 115, 144, 151, 238, 239  
 fourier (package) ..... 59  
 french (package) ..... 232  
 frenchpro (package) ..... 96, 232, 236

- G** garamondx (package) ..... 21, 146, 238  
 german (package) ..... 35, 231  
 \glb@currsiz ..... 1967, 3019
- H** hfoldsty (package) ..... 21, 144
- I** IEEEtran (package) ..... 99  
 \iffontchar ..... 731, 1329, 2818, 2830, 2831  
 \ifMT@auto ..... 304, 4553, 4614  
 \ifMT@babel ..... 304, 4700, 4828  
 \ifMT@do ..... 1085, 1152, 1789, 2279  
 \ifMT@document ..... 345, 3905  
 \ifMT@draft ..... 304, 4446  
 \ifMT@expansion ..... 304, 4502, 4516, 4611, 4657  
 \ifMT@fontspec ..... 850, 1065, 3148, 3568, 3594  
 \ifMT@ifo ..... 303, 867, 874, 897, 929, 2976, 4012, 4848, 4851  
 \ifMT@inannot ..... 99  
 \ifMT@inlist ..... 624, 669, 777, 807, 990, 1119, 1136, 1145, 1163, 1181, 1787, 2383, 2394, 2443, 2492, 2959, 2970, 2982, 3047, 3059, 3400  
 \ifMT@kerning ..... 304, 4714, 4841  
 \ifMT@nofamily ..... 1191, 1198
- J** jurabib (package) ..... 99
- K** kerning (option) ..... 6, 11, 31, 126  
 keyval (package) ..... 42, 63, 147, 236  
 \knaccode ..... 1738, 1739, 1749, 1752, 1758
- L** ledmac (package) ..... 29, 32, 53, 99, 233  
 ledpar (package) ..... 29  
 \leftmarginkern ..... 820  
 letterspace (option) ..... 8, 17, 24, 32, 45, 75, 128, 235  
 letterspace (package) ..... 1, 24, 31, 32, 34, 36, 38, 54, 76, 77, 79, 235–237, 239, 240  
 \letterspacefont ..... 1837  
 lineno (package) ..... 53  
 listings (package) ..... 26, 55, 56, 68, 88, 234, 235, 238  
 lmodern (package) ..... 8, 160  
 \LoadMicrotypeFile ..... 22, 44, 74, 3396, 4451  
 \loop ..... 3235  
 \lpcode ..... 1220, 1256, 1257, 1461, 1464, 1864
- M** marvosym (package) ..... 21, 32, 34, 151, 194, 234, 235  
 mathastext (package) ..... 54, 238  
 mathdesign (package) ..... 21, 146, 234  
 memoir (package) ..... 23, 99  
 \MicrotypeHook ..... 130, 4280  
 \microtypecontext ..... 22, 47, 918, 3008, 3027, 4453, 4811, 4814  
 microtypecontext (environment) ..... 22  
 \microtypesetup ..... 9, 46, 4284, 4452  
 MiKTeX (distribution) ..... 28  
 miniltx (package) ..... 24, 31, 34, 38, 52, 236  
 minimal (package) ..... 52, 56, 239  
 MinionPro (package) ..... 107  
 mmodern (package) ..... 21, 144, 240  
 \MT@font ..... 406, 993, 1016, 1055, 1057, 1319, 1333, 1801–1803, 1805, 1808, 2446, 2518, 2929  
 \MT@abbr@ex ..... 753  
 \MT@abbr@ex@e ..... 753  
 \MT@abbr@ex@inh ..... 753  
 \MT@abbr@kn ..... 753  
 \MT@abbr@kn@e ..... 753  
 \MT@abbr@kn@inh ..... 753  
 \MT@abbr@nl ..... 753  
 \MT@abbr@pr ..... 753  
 \glb@settings ..... 1968  
 \glet ..... 387  
 graphics (package) ..... 134  
 hyperref (package) ..... 54, 55, 99, 134, 233, 236  
 \ifMT@noligatures ..... 304, 4794  
 \ifMT@nonselected ..... 1517, 2505, 2513  
 \ifMT@norest ..... 2574, 2591, 2604, 2618, 2679  
 \ifMT@opt@auto ..... 4023, 4557, 4589  
 \ifMT@opt@DVI ..... 4023, 4471  
 \ifMT@opt@expansion ..... 4023, 4500, 4658  
 \ifMT@protrusion ..... 304, 814, 837, 4483, 4673  
 \ifMT@selected ..... 304, 4597, 4617  
 \ifMT@spacing ..... 304, 4682, 4699  
 \ifMT@tracking ..... 304, 4669, 4729  
 \ifMT@xunicode ..... 850, 2769, 2822  
 ifpdf (package) ..... 100, 134  
 \ignoringmicrotypeinpdfall ..... 105  
 \ignoreligaturesinfont ..... 2283  
 inputenc (package) ..... 14, 27, 32, 56, 68, 91, 95, 98, 233, 235, 236  
 \lslig ..... 24, 54, 1984, 1987  
 \lssyle ..... 23, 51, 920, 944, 1965, 2171, 2177, 2182, 2185, 2195, 2199, 4454  
 luaotfload (package) ..... 27, 30, 43, 50, 70, 76, 86, 93, 238, 239  
 LuaTeX (engine) ..... 1, 4, 6–9, 12–16, 20, 23–25, 27, 29–31, 34, 37–41, 43–45, 50, 53, 56, 58, 70, 74, 76, 83, 85, 86, 92, 117, 127, 133, 135, 136, 139, 144, 147, 151, 159, 236–240  
 luatexbase (package) ..... 43, 238–240  
 luatexja (package) ..... 27, 99, 100, 238, 239  
 luatextra (package) ..... 238  
 \MT@abbr@pr@e ..... 753  
 \MT@abbr@pr@inh ..... 753  
 \MT@abbr@sp ..... 753  
 \MT@abbr@sp@e ..... 753  
 \MT@abbr@sp@inh ..... 753  
 \MT@abbr@tr ..... 753  
 \MT@abbr@tr@e ..... 753  
 \MT@active@features ..... 1027, 2964, 2967, 2979, 2992, 3036, 3046, 3425, 4484, 4612, 4670, 4683, 4715, 4803, 4804  
 \MT@addto@annot ..... 99  
 \MT@addto@setup ..... 798, 799, 878, 1436, 2873, 3009, 3010, 3655, 3673, 4137, 4285, 4445, 4760, 4802, 4827, 4860  
 \MT@adjustspacing ..... 4345, 4346, 4613  
 \MT@afteraftergroup ..... 1893, 1901, 1910  
 \MT@auto ..... 1527, 4552, 4567, 4577, 4580, 4584  
 \MT@auto@ ..... 1527, 1550, 1595  
 \MT@autofalse ..... 306, 4565, 4576  
 \MT@autotru ..... 306, 4239, 4242, 4592  
 \MT@babelfalse ..... 313  
 \MT@babeltrue ..... 313  
 \MT@begin@catcodes ..... 2384, 2385, 2425, 3008, 3012, 3025,

- 3078, 3303, 3334, 3365, 3379, 3404, 3420,  
3454, 3470, 3492, 3513, 3527, 3792, 4255, 4256
- \MT@cat ..... 1432, 2864, 2866
- \MT@cfg@catcodes ..... 931, 1454, 2410, 2427
- \MT@char ..... 1245, 1246,  
1256, 1257, 1262, 1263, 1266, 1268, 1283,  
1284, 1287, 1461–1465, 1581, 1582, 1584,  
1585, 1623–1625, 1653, 1654, 1660, 1661,  
1667, 1668, 1671, 1672, 1677–1682, 1731,  
1732, 1738, 1739, 1742, 1743, 1748–1752,  
2291, 2292, 2294, 2547, 2563, 2571, 2573,  
2583, 2585, 2593, 2594, 2599, 2603, 2604,  
2609, 2611, 2617, 2619, 2622, 2626, 2630,  
2634, 2750, 2753, 2755, 2757, 2813, 2818,  
2820, 2823, 2825, 2831, 3844–3846, 3851–3853
- \MT@char@ ..... 1289, 1290, 1293, 1328,  
1329, 2547, 2554, 2558, 2563, 2612, 2683,  
2685, 2691, 2692, 2694, 2707, 2708, 2711,  
2712, 2715, 2716, 2720, 2722, 2754, 2756,  
2776, 2779, 2783, 2789, 2796, 2798, 2821, 2838
- \MT@charstring ..... 2560, 2753, 2761, 2788
- \MT@charxstring ..... 2791, 2825
- \MT@check@active@set .....  
..... 4266, 4490, 4630, 4672, 4692, 4719
- \MT@check@font ..... 989, 2955, 3038
- \MT@check@font@cx ..... 2965, 3038
- \MT@check@range ..... 3983, 3985
- \MT@check@range@ ..... 3985, 3986
- \MT@check@rlist ..... 3932, 3976
- \MT@check@rlist@ ..... 3976, 3977
- \MT@check@step ..... 4618
- \MT@checklist@ ..... 1093, 1112, 2276
- \MT@checklist@family ..... 1130
- \MT@checklist@font ..... 1174
- \MT@checklist@size ..... 1158
- \MT@checksetup ..... 4296, 4307, 4314, 4333, 4369
- \MT@clear@options ..... 163, 296, 788, 4228, 4438
- \MT@clist@break ..... 596, 1125, 1153, 1169, 1183
- \MT@clist@function ..... 596
- \MT@cnt@encoding ..... 3871, 3879, 3880
- \MT@cnt@family ..... 3877, 3886, 3887
- \MT@cnt@series ..... 3884, 3893, 3894
- \MT@cnt@shape ..... 3891, 3899, 3900
- \MT@config@file 4244, 4253, 4254, 4258, 4259, 4262
- \MT@context ..... 2468, 2480, 2499, 2526
- \MT@copy@font ..... 976,  
1012, 3651, 3657, 3669, 3676, 4146, 4148, 4474
- \MT@copy@font@ . 1012, 3651, 3657, 3669, 3676, 4146
- \MT@count ..... 512, 513,  
737, 1275, 1283, 1285, 1287, 1291, 1293,  
1296, 1302, 1307, 1308, 1312, 1323, 1355, 1417
- \MT@curr@file ..... 2388, 2389, 2398, 2399,  
3406, 3407, 3625, 3627, 3630, 3816, 4258, 4862
- \MT@curr@list@name ... 1321, 1358, 1448, 1591,  
2288, 2369, 2537, 2833, 2841, 2846, 2854, 2860
- \MT@curr@ls ..... 1872, 1988, 1991
- \MT@curr@set@name ..... 3102, 3104–  
3106, 3108, 3111, 3112, 3117, 3122, 3126,  
3127, 3136, 3161, 3165, 3194, 3222, 3245,  
3251, 3255, 3256, 3507, 3583, 3604, 3619,  
3638, 3710, 3714, 3721, 3724, 3726, 3732,  
3735, 3738, 3750, 3764, 3773, 3781, 3817, 3819
- \MT@declare@char@inh ..... 3798, 3806, 3814
- \MT@declare@sets ..... 3085, 3090, 3103, 3427
- \MT@DeclareMicrotypeAlias ..... 3380, 3382
- \MT@DeclareMicrotypeSetDefault ..... 3335, 3337
- \MT@DeclareSet ..... 3081, 3083, 3099
- \MT@DeclareSetAndUseIt ..... 3080, 3098
- \MT@DeclareVariants ..... 3367, 3368, 3370
- \MT@def@bool@opt ..... 4100,  
4113, 4114, 4118, 4132, 4144, 4154, 4166
- \MT@def@n ..... 393, 3073, 3074
- \MT@default@ex@set ..... 3352
- \MT@default@kn@set ..... 3352
- \MT@default@pr@set ..... 3352
- \MT@default@sp@set ..... 3352
- \MT@default@tr@set ..... 3352
- \MT@define@code@key .....  
... 3551, 3611, 3613, 3614, 3826, 3828, 3829
- \MT@define@code@key@family ..... 3562, 3612, 3827
- \MT@define@code@key@font ..... 3588, 3616, 3831
- \MT@define@code@key@size ..... 3576, 3615, 3830
- \MT@define@opt@key .. 3617, 3640–3643, 3777–3779
- \MT@define@optionX ..... 4284, 4342, 4345
- \MT@define@optionX@ ..... 4359, 4388, 4390, 4391
- \MT@define@set@key@ ..... 3115, 3295–3298
- \MT@define@set@key@font ..... 3243, 3300
- \MT@define@set@key@size ..... 3154, 3299
- \MT@detokenize@c ..... 436, 2555, 2751
- \MT@detokenize@n ..... 436, 2815
- \MT@dimen@six .....  
... 1048, 1276, 1312, 1355, 2070, 2072, 2241
- \MT@dinfo ..... 87
- \MT@dinfo@list .....  
1110, 1120, 1123, 1128, 1137, 1140, 1146,  
1148, 1156, 1164, 1167, 1172, 1182, 1185, 1189
- \MT@dinfo@nl ..... 87
- \MT@DisableLigatures ..... 3417
- \MT@do@font .... 704, 1222, 1555, 1690, 1759, 1864
- \MT@documentfalse ..... 345
- \MT@documenttrue ..... 345, 4806
- \MT@dofalse ... 1085, 1097, 1124, 1141, 1168, 1186
- \MT@dofont@function ..... 708, 719
- \MT@dotrue 1085, 1088, 1121, 1138, 1147, 1165, 2271
- \MT@draftfalse ..... 309, 4168
- \MT@drafttrue ..... 309, 4170
- \MT@edef@n ..... 395, 2504, 2530, 3063,  
3557, 3571, 3627, 3630, 3633, 3815, 3963, 4200
- \MT@encoding .... 1062, 1178, 1199, 1201, 1204,  
2463, 2475, 2555, 2751, 2814, 2845, 2852, 2860
- \MT@end@catcodes ..... 2386, 2429, 3009, 3016,  
3026, 3096, 3319, 3350, 3376, 3394, 3412,  
3429, 3465, 3489, 3508, 3524, 3538, 3809, 4257
- \MT@endinput ..... 167, 4223, 4232
- \MT@engine ..... 175,  
277, 281, 287, 293, 2172, 2176, 4224, 4230
- \MT@engine@toold ..... 175, 278
- \MT@error ..... 76, 780, 881,  
2172, 2183, 2361, 2374, 3192, 3328, 3357,  
3435, 3660, 3686, 3693, 4156, 4319, 4337,  
4377, 4558, 4570, 4590, 4659, 4730, 4743, 4823
- \MT@error@doesnt@work ..... 4726
- \MT@ex@c@name ..... 1509,  
1511, 1597, 1598, 1609, 1615, 1616, 1628, 3473
- \MT@ex@context ..... 1016, 3030, 3072
- \MT@ex@doc@contexts ..... 3072
- \MT@ex@factor ..... 314, 1528, 1601
- \MT@ex@factor@ . 1528, 1557, 1562, 1571, 1572, 1595
- \MT@ex@inh@name ..... 1583–1585
- \MT@ex@level ..... 314, 4613, 4615

- \MT@ex@max ..... 326, 1574, 1575
- \MT@ex@min ..... 326, 1577, 1578
- \MT@ex@setname ..... 3321
- \MT@ex@split@val ..... 1569
- \MT@exp@cs ..... 390, 393, 396, 399, 404, 1028,  
1030, 1039, 1162, 1267, 1342, 1345, 1375,  
1379, 1389, 1392, 1395, 1585, 1672, 1743,  
1913–1915, 2491, 2983, 2993, 2994, 2998,  
3049, 3060, 3160, 3582, 3602, 3853, 3934, 3981
- \MT@exp@gcs ..... 390, 394, 398, 400, 405
- \MT@exp@one@n ..... 407, 441, 603, 980,  
982, 1040, 1116, 1179, 1786, 2393, 2955,  
2958, 2968, 2980, 2986, 3004, 3046, 3057,  
3072, 3090, 3294, 3313, 3344, 3399, 3806, 4811
- \MT@exp@two@c .... 409, 427, 432, 442, 638, 979,  
1020, 1022, 1024, 1025, 1034, 1832, 1964,  
2000, 2003, 2006, 2552, 2559, 2752, 2823, 2824
- \MT@exp@two@n ..... 411, 1134, 1143, 3121, 3250
- \MT@expandfont ..... 1513, 1531, 1535
- \MT@expansion ..... 996, 1500, 4650
- \MT@expansionfalse ..... 305, 4508, 4513, 4607
- \MT@expansiontrue ..... 305, 4238
- \MT@extra@context .... 3072, 3457, 3473, 3496,  
3516, 3530, 3597, 3600, 3601, 3603, 3644,  
3652, 3658, 3670, 3675, 3788, 3931, 3935,  
3938, 3941, 3942, 3947, 3952, 3953, 3955, 3982
- \MT@extra@inputenc ..... 3786, 3813, 3818, 3819
- \MT@factor@default ..... 338, 4207, 4487
- \MT@family ..... 980, 1062,  
1135, 1178, 1199, 1201, 1204, 2464, 3390, 3391
- \MT@familyalias .....  
981, 982, 1074, 1142, 1144, 2474, 2476, 3392
- \MT@feat ..... 1085, 1248, 1337, 1339, 1341,  
1342, 1344, 1345, 1351, 1353, 1356, 1364–  
1368, 1370, 1372–1376, 1379, 1380, 1385,  
1389, 1392, 1395, 1398–1400, 1409, 1410,  
1415, 1433, 1807, 1969, 2359, 2361, 2364,  
2365, 2369, 2371, 2374, 2500, 2502, 2504,  
2508, 2512, 2516, 2517, 2520, 2528, 2530,  
2531, 2533, 2540, 2544, 2834, 2835, 2864, 2866
- \MT@features .....  
773, 3072, 3085, 3294, 3308, 3339, 3622, 3797
- \MT@features@long ..... 773, 776, 781, 3043, 3811
- \MT@file@list ..... 2380,  
2382, 2390, 2393, 2395, 2400, 2403, 3399, 3403
- \MT@find@file ..... 980, 982, 2380
- \MT@fix@catcode ..... 5
- \MT@fix@font@set ..... 3144, 4458
- \MT@fix@fontdimen@six ..... 1048, 1073
- \MT@font ..... 406, 731, 733,  
973, 979, 1034, 1041, 1050–1053, 1056,  
1059, 1220, 1221, 1256, 1257, 1262, 1263,  
1283, 1284, 1287, 1290, 1293, 1318, 1323,  
1329, 1423, 1426, 1461, 1462, 1464, 1465,  
1540, 1544, 1550, 1555, 1581, 1582, 1623,  
1625, 1653, 1654, 1660, 1661, 1667, 1668,  
1677–1679, 1681, 1682, 1687–1689, 1731,  
1732, 1738, 1739, 1748, 1749, 1751, 1752,  
1757, 1758, 1786, 1790, 1870, 1871, 2280,  
2570, 2625, 2868, 2924–2926, 2928, 2943,  
2944, 2955, 2956, 2961, 2968, 2986, 2996, 3005
- \MT@font@copy . 1016, 1024, 1026, 1034–1036, 1046
- \MT@font@list .....  
.. 2868, 2955, 2956, 2958, 2960, 2961, 3037
- \MT@font@orig ..... 1018
- \MT@font@sets ..... 3125, 3144, 3254, 4458
- \MT@fontspecfalse ..... 852
- \MT@fontspectrue ..... 852, 853, 891
- \MT@gdef@n ..... 393, 3448, 3464, 3488, 3523,  
3537, 3726, 3750, 3764, 3821, 3864–3869, 4274
- \MT@get@axis ..... 3268–3271, 3281
- \MT@get@basefamily ..... 2392, 2430
- \MT@get@basefamily@ ..... 2436, 2439
- \MT@get@char@unit ..... 1247, 1387, 1413, 1505
- \MT@get@charwd ..... 1281, 1312, 1390, 1413
- \MT@get@config ..... 4244
- \MT@get@ex@opt ..... 1504, 1522, 1595
- \MT@get@ex@opt@ ..... 1603–1605, 1607, 1614
- \MT@get@font ..... 3249, 3259
- \MT@get@font@ ..... 3260, 3265, 3609
- \MT@get@font@and@size ..... 3593, 3608
- \MT@get@font@dimen ..... 1316, 1393
- \MT@get@font@highlevel ... 3120, 3130, 3283, 3556, 3567
- \MT@get@inh@list .... 1209, 1507, 1640, 1718, 2526
- \MT@get@listname ..... 2445, 2502, 2528
- \MT@get@listname@ ..... 2445
- \MT@get@ls@basefont ..... 1825, 1990, 1997
- \MT@get@opt ..... 1207, 1362, 1638, 1716
- \MT@get@range ..... 3158, 3168, 3580
- \MT@get@size ..... 3175, 3180, 3189, 3208, 3279
- \MT@get@size@ ..... 3215, 3226
- \MT@get@size@@ ..... 3226
- \MT@get@slot ..... 1244, 2290, 2547, 3843, 3850
- \MT@get@slot@ ..... 2564, 2568
- \MT@get@space@unit .....  
.. 1387, 1489, 1651, 1658, 1665, 1729, 1736
- \MT@get@tr@opt ..... 1812, 1932
- \MT@get@tr@opt@ ..... 1952–1955, 1957
- \MT@get@unit ..... 1395, 1403, 1941
- \MT@get@unit@ ..... 1403
- \MT@getkey ..... 4425, 4440
- \MT@gl@et . 387, 400, 892, 1020, 1914, 2000, 2031,  
2177, 2185, 2922, 3019, 3038–3041, 3392,  
3651, 3669, 3874, 4146, 4148, 4774, 4780, 4866
- \MT@gl@et@nc .....  
399, 1057, 1204, 1808, 2044, 2232, 2540,  
2944, 3037, 3108, 3111, 3117, 3245, 3481,  
3710, 3714, 3721, 3724, 3732, 3735, 3738, 3773
- \MT@gl@et@nn ..... 404, 3056, 4271
- \MT@if@false ..... 303,  
863, 870, 893, 910, 2973, 3980, 4842, 4849
- \MT@if@list@exists .....  
.. 1197, 1503, 1521, 1637, 1715, 1811, 2499
- \MT@if@luaotf@font .....  
.... 687, 707, 1827, 1862, 1871, 2293, 2306
- \MT@if@outer@next .....  
... 2111, 2126, 2130, 2133, 2138, 2143–2146
- \MT@if@true ..... 303,  
864–866, 871–873, 894–896, 923, 927, 928,  
2966, 3989, 3993, 4001, 4006, 4843–4847, 4850
- \MT@ifdefined@c@T ..... 415, 867, 899,  
1265, 1301, 1583, 1670, 1741, 1856, 1946,  
2738, 3146, 3390, 3818, 3921, 4280, 4283, 4869
- \MT@ifdefined@c@TF .....  
..... 415, 815, 1815, 1840, 1843, 1854,  
1988, 2067, 2238, 2285, 2286, 2503, 2529, 3930
- \MT@ifdefined@n@T .....  
.. 415, 804, 1113, 1131, 1159, 1175, 1266,  
1398, 1434, 1584, 1609, 1671, 1742, 1936,

- 1944, 1958, 2482, 2863, 3105, 3385, 3476,  
3634, 3880, 3887, 3894, 3900, 3931, 3941, 4425
- `\MT@ifdefined@n@TF` ..... 415, 1074,  
1091, 1114, 1132, 1160, 1176, 1199, 1364,  
1372, 1597, 1615, 1802, 1912, 2274, 2364,  
2470, 2555, 3133, 3322, 3325, 3353, 3625,  
3907, 3962, 3971, 4041, 4267, 4270, 4631, 4809
- `\MT@ifdim` .. 535, 658, 659, 663, 664, 3191, 3200,  
3987–3989, 3991, 3992, 3999–4001, 4004, 4005
- `\MT@ifdimen` .... 497, 3217, 3713, 3723, 3737, 4216
- `\MT@ifempty` ..... 448, 1220, 1221, 1254, 1260,  
1478, 1479, 1650, 1657, 1664, 1687–1689,  
1701–1703, 1705–1707, 1728, 1735, 1757,  
1758, 1774, 1775, 2050, 2051, 2059, 2075,  
2207, 2230, 2231, 2243, 3048, 3084, 3088,  
3132, 3142, 3169, 3170, 3183, 3184, 3273,  
3284, 3307, 3311, 3338, 3342, 3424, 3503,  
3618, 3624, 3644, 3650, 3658, 3668, 3675,  
3748, 3796, 3803, 4054, 4085, 4247, 4293, 4366
- `\MT@ifint` ..... 461, 2714, 3504, 3749, 4199, 4209
- `\MT@ifstreq` . 544, 1103, 1440, 1607, 2360, 2501,  
2516, 2777, 2780, 2788, 2797, 3003, 3248,  
3592, 3712, 3734, 3736, 3798, 3805, 3922,  
3942, 4013, 4057, 4059, 4062, 4065, 4087,  
4089, 4103, 4104, 4176, 4177, 4182, 4186,  
4215, 4295, 4302, 4306, 4313, 4368, 4374, 4702
- `\MT@in@clist` ..... 624,  
776, 805, 1116, 1134, 1143, 1179, 1786,  
2382, 2393, 2955, 2958, 2968, 2980, 3046, 3399
- `\MT@in@rlist` ..... 652, 1162, 2491
- `\MT@in@rlist@` ..... 652
- `\MT@in@rlist@` ..... 652
- `\MT@in@tlist` ..... 640, 2442, 3057
- `\MT@in@tlist@` ..... 640
- `\MT@inannotfalse` ..... 100
- `\MT@inannottrue` ..... 100
- `\MT@increment` ..... 737, 3879, 3886, 3893, 3899
- `\MT@info` ..... 76, 91, 96, 1055, 4183, 4409, 4413
- `\MT@info@missing@char` ..... 1296, 1326, 2621
- `\MT@info@nl` ..... 76, 92, 93, 97, 816, 1200,  
1327, 4174, 4184, 4254, 4268, 4272, 4470,  
4474, 4486, 4493, 4614, 4651, 4671, 4676,  
4685, 4695, 4718, 4722, 4735, 4750, 4819, 4830
- `\MT@info@nottracking` ..... 994, 1104, 1108
- `\MT@info@nottracking@` ..... 1104, 1108
- `\MT@inh@do` ..... 2539, 3832
- `\MT@inh@feat` ..... 3786, 3796, 3801, 3812
- `\MT@inh@split` ..... 3834, 3838
- `\MT@inlist@false` ..... 624, 628, 641, 653, 2976
- `\MT@inlist@true` ..... 624, 630, 648, 660, 665, 2976
- `\MT@is@active` ..... 2552, 2727
- `\MT@is@char` ..... 2559, 2752, 2761
- `\MT@is@charx` ..... 2770, 2791, 2824
- `\MT@is@composite` ..... 2557, 2811
- `\MT@is@feature` ..... 775, 3089, 3312, 3343
- `\MT@is@letter` ..... 2553, 2671, 2757, 2820
- `\MT@is@number` ..... 2697, 2702, 2808
- `\MT@is@symbol` ..... 2556, 2749
- `\MT@is@tlig` ..... 2755, 2805
- `\MT@is@uni@comp` ..... 2818, 2830
- `\MT@is@xchar` ..... 2781, 2787
- `\MT@iterate` ..... 674
- `\MT@kerning` ..... 1000, 1712, 4721
- `\MT@kerningfalse` ..... 311
- `\MT@kerningtrue` ..... 311
- `\MT@kn@c@name` ..... 1720, 1722, 1765, 3530
- `\MT@kn@context` ..... 3031, 3072
- `\MT@kn@doc@contexts` ..... 3072
- `\MT@kn@factor` ..... 314
- `\MT@kn@factor@` ..... 1364
- `\MT@kn@inh@name` ..... 1741–1743
- `\MT@kn@max` ..... 326
- `\MT@kn@min` ..... 326
- `\MT@kn@setname` ..... 3321
- `\MT@kn@split@val` ..... 1726
- `\MT@kn@unit` ..... 320
- `\MT@kn@unit@` ..... 1372, 1768
- `\MT@led@unhbox@line` ..... 816
- `\MT@ledmac@setup` ..... 810, 887–889
- `\MT@let@cn` ..... 401, 404, 405, 1075, 1213,  
1511, 1598, 1644, 1722, 1937, 1945, 2359,  
2371, 2496, 2500, 2533, 2924, 3212, 3638, 3817
- `\MT@let@enc` .. 399, 1400, 2152, 2512, 2544, 2971,  
3964, 3969, 4063, 4066, 4309, 4316, 4733, 4748
- `\MT@let@nn` ..... 404, 1365, 1370, 1373, 1385, 1616, 1619, 1959
- `\MT@letterspace` ..... 341, 1815, 1945, 1947, 1949, 4783, 4784, 4786
- `\MT@letterspace@` ..... 1231, 1301,  
1302, 1815, 1816, 1819, 1822, 1830, 1837,  
1842, 1850, 1863–1865, 1872, 1881, 1925–  
1928, 2072, 2075, 2193, 2208–2211, 2244, 2247
- `\MT@letterspace@default` ..... 341, 4784
- `\MT@listname` ..... 2445,  
2493, 2496, 2503, 2504, 2506, 2508, 2529,  
2530, 2532, 2533, 2537, 2540, 3853, 3858
- `\MT@listname@count` ..... 3621, 3626, 3628
- `\MT@load@inputenc` ..... 1441, 1444, 1453
- `\MT@load@list` ..... 1211, 1509, 1642, 1720, 2356
- `\MT@loop` ..... 674, 682, 730, 3968
- `\MT@lower` ..... 3162, 3168, 3583
- `\MT@ls@adjust` ..... 1879, 2204
- `\MT@ls@adjust@` ..... 2188, 2212
- `\MT@ls@adjust@empty` ..... 2190, 2204
- `\MT@ls@adjust@relax` ..... 2191, 2204
- `\MT@ls@basefont` ..... 1997, 2006, 2009, 2010
- `\MT@ls@fontspec@colon` ..... 1921
- `\MT@ls@fontspec@font` ..... 1832, 1921
- `\MT@ls@outer@k` ..... 1882,  
1887, 1895, 2088, 2102, 2149, 2167, 2252
- `\MT@ls@set@ls` ..... 2195, 2198, 2204
- `\MT@ls@too@large` ..... 2211, 2214, 4786
- `\MT@ls@font` ..... 1821,  
1831, 1837, 1850, 1851, 1860, 1864, 1865,  
1870, 1964, 1975, 2020, 2022, 2037, 2038,  
2045, 2046, 2053, 2055, 2061, 2076, 2245, 2248
- `\MT@ltx@pickupfont` ..... 916, 957, 962, 2939, 2947
- `\MT@lua` ..... 192, 255, 356,  
364, 485, 522, 568, 688, 709, 2294, 2307, 2575
- `\MT@lua@copy@font` ..... 1045
- `\MT@lua@copyfont` ..... 1025, 1045
- `\MT@luatex@no` ..... 253, 354
- `\MT@map@clist@` ..... 596
- `\MT@map@clist@c` ..... 596,  
1027, 2289, 2967, 2979, 2992, 3036, 3043,  
3085, 3308, 3339, 3622, 3797, 3801, 3811, 4423
- `\MT@map@clist@n` ..... 596, 1090, 2273, 3072, 3087, 3107, 3118,  
3156, 3246, 3294, 3310, 3341, 3371, 3446,



- 3554, 3565, 3578, 3590, 3730, 3746, 3847,  
 4049, 4052, 4081, 4083, 4112, 4194, 4291, 4364  
 \MT@map@tlist@ . . . . . 614  
 \MT@map@tlist@c . . . . . 614, 643, 654, 1030, 1267,  
 1585, 1672, 1743, 2449, 2998, 3932, 3981, 4458  
 \MT@map@tlist@n . . . . . 614, 2421, 2763, 3873  
 \MT@max@char . . . . . 2671, 2694  
 \MT@max@slot . . . . . 2671, 2720  
 \MT@maybe@do . . . . . 1085, 1194, 1501, 1635, 1713, 1788  
 \MT@maybe@gobble@with@tikz . . . . . 854, 1911, 1968  
 \MT@maybe@rem@from@list . . . . . 2998, 3002  
 \MT@microtype@context . . . . . 3008  
 \MT@minus . . . . . 1830, 1925–1928  
 \MT@MT . . . . . 2, 76, 79, 82, 84,  
 86, 91, 92, 94, 95, 292, 787, 882, 884, 3660,  
 4226, 4248, 4409, 4413, 4471, 4706, 4823, 4824  
 \MT@MT@pickupfont . . . . . 959, 964, 2939, 2949  
 \MT@next@listname . . . . . 2452, 2461  
 \MT@next@listname@ . . . . . 2471, 2483, 2489  
 \MT@n@ligatures . . . . . 2280, 3417  
 \MT@n@setname . . . . . 2272, 3417  
 \MT@nofamily@false . . . . . 1191, 1196  
 \MT@nofamily@true . . . . . 1191, 2462  
 \MT@n@ligatures . . . . . 1004, 2268, 4795  
 \MT@n@ligatures@ . . . . . 2020, 2022, 2280, 2283  
 \MT@n@ligatures@false . . . . . 308  
 \MT@n@ligatures@true . . . . . 308, 3426  
 \MT@nonselected@false . . . . . 1517, 1532  
 \MT@nonselected@true . . . . . 1517, 1520  
 \MT@norest@false . . . . .  
 . . . . . 2680, 2687, 2696, 2717, 2777, 2789, 2797  
 \MT@norest@true . . . . . 2550, 2680, 2705  
 \MT@old@cmd . . . . . 66, 71–74  
 \MT@opt@autofalse . . . . . 4024  
 \MT@opt@autotruer . . . . . 4024, 4114  
 \MT@opt@def@set . . . . . 4040, 4068, 4093  
 \MT@opt@DVIfalse . . . . . 4025  
 \MT@opt@DVIttrue . . . . . 4025, 4121, 4124  
 \MT@opt@expansion@false . . . . . 4023  
 \MT@opt@expansion@true . . . . . 4023  
 \MT@options . . . . . 3541  
 \MT@optwarn@admissible . . . . . 4026, 4105, 4186  
 \MT@optwarn@nan . . . . . 4030, 4201, 4211  
 \MT@orig@add@accent . . . . . 2945  
 \MT@orig@foreign@language . . . . . 4836, 4838  
 \MT@orig@pickupfont . . . . . 2874, 2941  
 \MT@orig@py@macron . . . . . 954, 958, 963  
 \MT@orig@select@language . . . . . 4831, 4833  
 \MT@outer@kern 1881, 1885, 1887, 1888, 1892, 1900,  
 1991, 1992, 1994, 2149, 2150, 2167, 2225, 2264  
 \MT@outer@space . . . . . 1875,  
 1876, 2035, 2084, 2086, 2087, 2100, 2101,  
 2118, 2119, 2134, 2135, 2139, 2140, 2165  
 \MT@pdf@annot . . . . . 99  
 \MT@pdf@or@lua . . . . . 180, 188, 290, 464, 687, 4767  
 \MT@pdf@tex@no . . . . . 212, 353  
 \MT@permute 3463, 3487, 3501, 3522, 3536, 3824, 3863  
 \MT@permute@ . . . . . 3863  
 \MT@permute@@ . . . . . 3863  
 \MT@permute@@@ . . . . . 3863  
 \MT@permute@@@@ . . . . . 3898, 3903  
 \MT@permute@@@@@ . . . . . 3911, 3913  
 \MT@permute@define . . . . . 3904, 3914–3916, 3960  
 \MT@permute@reset . . . . . 3873, 3966  
 \MT@permutelist . . . . . 3457, 3473,  
 3499, 3516, 3530, 3597, 3598, 3603, 3822,  
 3926, 3927, 3931, 3935, 3938, 3941–3943,  
 3945–3947, 3955, 3956, 3982, 4014, 4016, 4017  
 \MT@pickupfont . . . . . 2939  
 \MT@plain . . . . . 125, 146  
 \MT@pr@c@name . . . . . 1211, 1213, 1469, 3457  
 \MT@pr@context . . . . . 1016, 3030, 3072  
 \MT@pr@doc@contexts . . . . . 3072  
 \MT@pr@factor . . . . . 314, 4210, 4487, 4488  
 \MT@pr@factor@ . . . . . 1364  
 \MT@pr@inh@name . . . . . 1265, 1266, 1268  
 \MT@pr@level . . . . . 314, 4485, 4486  
 \MT@pr@max . . . . . 326  
 \MT@pr@min . . . . . 326  
 \MT@pr@setname . . . . . 3321  
 \MT@pr@split@val . . . . . 1252  
 \MT@pr@unit . . . . . 320, 4217, 4489  
 \MT@pr@unit@ . . . . . 1372, 1472  
 \MT@preset@aux . . . . . 1474,  
 1476, 1478, 1479, 1482, 1770, 1772, 1774, 1775  
 \MT@preset@aux@factor 1474, 1482, 1701–1703, 1770  
 \MT@preset@aux@space . 1476, 1482, 1705–1707, 1772  
 \MT@preset@ex . . . . . 1610, 1627  
 \MT@preset@kn . . . . . 1763  
 \MT@preset@kn@ . . . . . 1763  
 \MT@preset@pr . . . . . 1467  
 \MT@preset@pr@ . . . . . 1467  
 \MT@preset@sp . . . . . 1694  
 \MT@preset@sp@ . . . . . 1694  
 \MT@ProcessOptionsWithKV . . . . . 4417, 4441  
 \MT@protrude@chars . . . . . 4342, 4346, 4485, 4673  
 \MT@protrusion . . . . . 995, 1193, 4492  
 \MT@protrusion@false . . . . . 304  
 \MT@protrusion@true . . . . . 304, 4233  
 \MT@rbba@expansion . . . . . 768  
 \MT@rbba@kerning . . . . . 768  
 \MT@rbba@protrusion . . . . . 768  
 \MT@rbba@spacing . . . . . 768  
 \MT@rbba@tracking . . . . . 768  
 \MT@register@font . . . . . 1008, 2956, 3039  
 \MT@register@font@cx . . . . . 2991, 3039  
 \MT@register@subst@font . . . . . 2930, 2957, 3040  
 \MT@register@subst@font@cx . . . . . 2978, 3040  
 \MT@rem@from@c@list 635, 1040, 2961, 2986, 3004, 3546  
 \MT@rem@from@list . . . . . 1030, 1038  
 \MT@rem@last@space . . . . . 436  
 \MT@repeat . . . . . 674, 684, 733, 3974  
 \MT@requires@latex . . . . . 145, 164, 796,  
 912, 1963, 2872, 4032, 4193, 4418, 4422, 4865  
 \MT@requires@luatex 348, 387, 986, 1053, 1229,  
 1536, 1537, 1559, 1606, 1782, 2283, 3763,  
 3769, 4117, 4347, 4361, 4503, 4580, 4588, 4667  
 \MT@requires@pdftex . . . . . 348,  
 463, 498, 545, 811, 975, 985, 999, 1003,  
 1014, 1228, 1273, 1299, 1558, 1634, 1712,  
 1781, 2017, 2269, 3418, 3648, 3762, 3767,  
 4143, 4237, 4360, 4524, 4555, 4583, 4666, 4793  
 \MT@res@a . . . . . 558, 560, 626, 633, 636, 638, 642, 647  
 \MT@res@b . . . . . 559, 560, 636–638, 646, 647  
 \MT@reset@context . . . . . 3018, 3022, 3028, 3055  
 \MT@reset@context@ . . . . . 3028, 3055  
 \MT@reset@ef@codes . . . . . 1506, 1530, 1558, 1611  
 \MT@reset@ef@codes@ . . . . . 1553, 1563, 1567  
 \MT@reset@kn@codes . . . . . 1717, 1724, 1754

- \MT@reset@kn@codes@ ..... 1754  
 \MT@reset@pr@codes ..... 1208, 1215, 1224  
 \MT@reset@pr@codes@ ..... 1224  
 \MT@reset@sp@codes ..... 1639, 1646, 1684  
 \MT@reset@sp@codes@ ..... 1684  
 \MT@restore@catcodes ..... 5, 7, 8, 159, 165, 4871  
 \MT@restore@p@h ..... 847, 874, 929  
 \MT@saved@setupfont ..... 4406, 4414  
 \MT@scale ..... 744, 1276, 1338, 1352, 1355, 1572  
 \MT@scale@factor ..... 1278, 1309, 1336, 1484, 1629  
 \MT@scale@to@em .. 1255, 1261, 1273, 1306, 1490,  
     1652, 1659, 1666, 1730, 1737, 1948, 2068, 2239  
 \MT@scrubfeature ..... 1066, 1078, 3569  
 \MT@scrubfeatures ..... 1078, 3149, 3595  
 \MT@selectedfalse ..... 307  
 \MT@selectedtrue ..... 307  
 \MT@series ..... 1062, 1178, 2465, 2477  
 \MT@set@all@ex ..... 1553, 1630  
 \MT@set@all@kn ..... 1754, 1776  
 \MT@set@all@pr ..... 1217, 1224, 1480  
 \MT@set@all@sp ..... 1684, 1709  
 \MT@set@babel@context ..... 4808, 4834, 4839, 4853  
 \MT@set@codes . 1214, 1235, 1512, 1645, 1723, 2372  
 \MT@set@curr@ls ..... 1872  
 \MT@set@curr@ok 1892, 1894, 1900, 1903, 2123, 2127  
 \MT@set@curr@os ..... 1874, 1902, 2123, 2127  
 \MT@set@default@set ..... 3339, 3344, 3352  
 \MT@set@ex@codes ..... 1534, 4598, 4600  
 \MT@set@ex@codes@n ..... 1517, 1534, 4600  
 \MT@set@ex@codes@s ..... 1502, 4598  
 \MT@set@ex@heirs ..... 1585, 1622  
 \MT@set@inh@list ..... 3793, 3795  
 \MT@set@inputenc 1210, 1431, 1508, 1641, 1719, 2538  
 \MT@set@inputenc@ ..... 1434, 1436  
 \MT@set@kn@codes ..... 1714  
 \MT@set@kn@heirs ..... 1743, 1747  
 \MT@set@listname ..... 1212,  
     1363, 1510, 1596, 1643, 1721, 1934, 2833  
 \MT@set@lsbasefont ..... 2006  
 \MT@set@lsfont ..... 1869, 1962, 1975  
 \MT@set@named@keys 3459, 3475, 3497, 3518, 3532, 3541  
 \MT@set@pr@codes ..... 1195  
 \MT@set@pr@heirs ..... 1269, 1459  
 \MT@set@sp@codes ..... 1636  
 \MT@set@sp@heirs ..... 1672, 1676  
 \MT@set@tr@codes ..... 1798, 1970, 1980  
 \MT@set@tr@zero ..... 1817, 2006, 4388  
 \MT@SetExpansion ..... 3471, 3473  
 \MT@SetExtraKerning ..... 3528, 3530  
 \MT@SetExtraSpacing ..... 3514, 3516  
 \MT@SetProtrusion ..... 3455, 3457  
 \MT@SetTracking ..... 3493, 3495  
 \MT@setup@ ..... 794, 798, 4135, 4136, 4866  
 \MT@setup@contexts ..... 3017, 3035  
 \MT@setup@copies ..... 4457, 4473  
 \MT@setup@expansion ..... 4460, 4497  
 \MT@setup@kerning ..... 4464, 4713, 4756  
 \MT@setup@noligatures ..... 4465, 4791  
 \MT@setup@PDF ..... 4456, 4468, 4478  
 \MT@setup@protrusion ..... 4459, 4481  
 \MT@setup@spacing ..... 4463, 4680, 4757  
 \MT@setup@spacing@check ..... 4698, 4869, 4870  
 \MT@setup@tracking ..... 4461, 4665, 4728, 4755  
 \MT@setup@warntracking ..... 4462, 4760  
 \MT@setupfont .. 971, 2933, 4407, 4410, 4414, 4450  
 \MT@setupfont@hook 862, 892, 898, 903, 929, 936, 974  
 \MT@shape ..... 1062, 1178, 2466, 2478  
 \MT@shorthandoff ..... 4817, 4848, 4851  
 \MT@show@pdfannot ..... 106  
 \MT@shrink ..... 323, 1525,  
     4520, 4521, 4528, 4529, 4532, 4536, 4603, 4616  
 \MT@shrink@ ..... 1525, 1540, 1544, 1550, 1595  
 \MT@shrink@default ..... 339  
 \MT@size ..... 659, 663, 664, 1054, 1062, 1178  
 \MT@size@name ..... 652, 2493  
 \MT@SOUL@doword ..... 947, 948  
 \MT@sp@cname ..... 1642, 1644, 1696, 3516  
 \MT@sp@context ..... 3031, 3072, 4702  
 \MT@sp@doc@contexts ..... 3072  
 \MT@sp@factor ..... 314  
 \MT@sp@factor@ ..... 1364  
 \MT@sp@inh@name ..... 1670–1672  
 \MT@sp@max ..... 326  
 \MT@sp@min ..... 326  
 \MT@sp@setname ..... 3321  
 \MT@sp@split@val ..... 1648  
 \MT@sp@unit ..... 320  
 \MT@sp@unit@ ..... 1372, 1699  
 \MT@spacing ..... 1000, 1633, 4694  
 \MT@spacingfalse ..... 310  
 \MT@spacingtrue ..... 310  
 \MT@split@codes ..... 1237, 1241  
 \MT@split@name ..... 979, 1062  
 \MT@step ..... 323, 1526, 4523,  
     4525, 4543, 4547, 4550, 4617, 4620, 4621, 4625  
 \MT@step@ ..... 1526, 1540, 1544, 1550, 1595  
 \MT@stretch ..... 323, 1524, 4517,  
     4518, 4521, 4528, 4530, 4535, 4538, 4602, 4616  
 \MT@stretch@ ..... 1524, 1540, 1544, 1550, 1595  
 \MT@stretch@default ..... 339, 4518  
 \MT@strip@prefix ..... 2770, 2791, 2823  
 \MT@temp ... 1219–1222, 1686–1690, 1756–1759,  
     2040, 2044, 2047, 2055, 2058, 2115, 2118,  
     2124, 2128, 2131, 2134, 2139, 2147, 2149,  
     2155, 2229, 2232, 2249, 3267, 3292, 4421,  
     4426, 4431, 4432, 4434, 4437, 4632, 4636, 4640  
 \MT@tempencoding ..... 3906, 3907, 3917, 3923  
 \MT@tempfamily ..... 3918  
 \MT@tempseries ..... 3919  
 \MT@tempshape ..... 3920  
 \MT@tempsize ..... 3874, 3921, 3930, 3932, 3936  
 \MT@test@ast ..... 2066, 2237, 3131, 3140  
 \MT@text@microtypecontext ..... 3025  
 \MT@textls ..... 49, 2190, 2191, 2193  
 \MT@textmicrotypecontext ..... 3025  
 \MT@the@pr@code ..... 1226, 1256, 1262, 1866  
 \MT@the@pr@code@tr ..... 1226, 1866  
 \MT@tikz@setup ..... 854, 875, 952  
 \MT@tlist@break ..... 614, 649, 671, 2454, 4020  
 \MT@toks ..... 302,  
     1327, 1358, 1591, 2551, 2586, 2613, 2685,  
     2692, 2742, 2783, 2798, 2807, 2838, 2850, 2858  
 \MT@tr@cname ..... 1936,  
     1937, 1944, 1945, 1958, 1959, 3498, 3505  
 \MT@tr@context ..... 3031, 3072  
 \MT@tr@doc@contexts ..... 3072  
 \MT@tr@factor@ ..... 1935  
 \MT@tr@font@list ..... 1784  
 \MT@tr@ispace ..... 1840, 1841, 1952  
 \MT@tr@ligatures ..... 1856, 1955, 2019, 2022

- `\MT@tr@max` ..... 326, 2215–2217  
`\MT@tr@min` ..... 326, 2219–2221  
`\MT@tr@noligatures` ..... 1856, 2016  
`\MT@tr@okern` ..... 1854, 1954  
`\MT@tr@ospace` ..... 1843, 1844, 1952  
`\MT@tr@outer@icr` ..... 2128, 2157  
`\MT@tr@outer@icr@` ..... 2157  
`\MT@tr@outer@l` ..... 1877, 2080  
`\MT@tr@outer@next` ..... 2108, 2112, 2159  
`\MT@tr@outer@r` ..... 1904, 2108, 2124, 2131, 2157  
`\MT@tr@outer@r@` ..... 2109, 2114  
`\MT@tr@set@okern` ..... 1855, 2225  
`\MT@tr@set@okern@` ..... 2230, 2231, 2236  
`\MT@tr@set@space` ..... 1846, 2036  
`\MT@tr@set@space@` ..... 2041–2043, 2049  
`\MT@tr@set@space@@` ..... 2052, 2057, 2060, 2065  
`\MT@tr@setname` ..... 3321  
`\MT@tr@unit@` ..... 1936, 2067, 2238  
`\MT@tracking` 987, 1784, 1970, 2935, 4388, 4389, 4675  
`\MT@tracking@` ..... 1784, 4389  
`\MT@trackingfalse` ..... 312, 4732  
`\MT@trackingtrue` ..... 312  
`\MT@try@order` ..... 2449, 2457  
`\MT@undefined@char` ..... 2732, 2737, 2748  
`\MT@upper` ..... 3162, 3168, 3583  
`\MT@use@set` ..... 3308, 3313, 3321  
`\MT@UseMicrotypeSet` ..... 3304, 3306  
`\MT@val` ..... 3048–3050, 3052, 3058, 3060,  
3063, 3119, 3123, 3131, 3134, 3137, 3157–  
3159, 3171, 3174, 3176, 3179, 3181, 3182,  
3188, 3190, 3191, 3193, 3196, 3198, 3209,  
3212, 3217–3219, 3221, 3223, 3228, 3247–  
3249, 3252, 3261–3263, 3275, 3277, 3282,  
3284, 3286, 3288, 3555, 3557, 3566, 3569,  
3571, 3579–3581, 3591–3593, 3599, 3604,  
3846, 3853, 3857, 3858, 4041, 4042, 4045,  
4053, 4054, 4057, 4059, 4062, 4065, 4084,  
4085, 4087, 4089, 4292, 4293, 4295, 4302,  
4306, 4313, 4319, 4365, 4366, 4368, 4374, 4377  
`\MT@variants` ..... 2442, 3363, 3374  
`\MT@vinfo` ..... 76, 93, 993, 1105, 1109, 1367,  
1376, 1380, 1415, 1599, 1617, 1801, 1819,  
2297, 2308, 2365, 2389, 2399, 2402, 2506,  
2508, 2514, 3029, 3052, 3401, 3407, 4174,  
4181, 4298, 4304, 4310, 4317, 4371, 4376, 4810  
`\MT@warn@ascii` ..... 2694, 2837  
`\MT@warn@axis@empty` ..... 3274, 3285, 3290  
`\MT@warn@code@too@large` ..... 1342, 1345, 1349  
`\MT@warn@err` ..... 76, 4178, 4179  
`\MT@warn@ex@too@large` ..... 1575, 1578, 1589  
`\MT@warn@maybe@inputenc` ... 2287, 2851, 2859, 2862  
`\MT@warn@nodim` ..... 3702, 3715, 3725, 3739  
`\MT@warn@number@too@large` ..... 2721, 2843  
`\MT@warn@preset@twidth` ... 1473, 1493, 1700, 1769  
`\MT@warn@rest` ..... 2592, 2629, 2848  
`\MT@warn@tracking@DVI` ..... 1820, 4762  
`\MT@warn@unknown` .... 2582, 2598, 2608, 2633, 2856  
`\MT@warn@unknown@once` ..... 2874, 2875  
`\MT@warning` . 67, 76, 94, 2216, 2220, 2300, 2517,  
3106, 3135, 3221, 3291, 3386, 3409, 3506,  
3635, 3704, 3752, 3765, 3770, 3924, 3945,  
4016, 4178, 4183, 4280, 4357, 4396, 4398, 4399  
`\MT@warning@enl` 76, 88, 95, 158, 279, 786, 790, 826,  
838, 905, 1319, 1356, 1408, 1448, 1494,  
1590, 1804, 2028, 2838, 2844, 2849, 2857,  
2906, 3478, 3677, 4027, 4034, 4045, 4127,  
4179, 4184, 4218, 4225, 4261, 4275, 4447,  
4504, 4548, 4604, 4623, 4687, 4703, 4768, 4855  
`\MT@while@num` ..... 680, 704, 710, 932–934  
`\MT@with@babel@and@T` .....  
... 803, 864–866, 894–896, 4843–4847, 4850  
`\MT@with@package@T` ..... 800, 851, 853,  
868, 871–873, 875, 880, 887–891, 901, 913,  
927, 928, 930, 952, 953, 2874, 2875, 2917, 4686  
`\MT@xadd` ..... 104, 580, 2390, 2395, 2400,  
2403, 2983, 2994, 3060, 3160, 3403, 3582, 3853  
`\MT@xaddb` ..... 588, 3602, 3934  
`\MT@xdef@n` .. 395, 3323, 3326, 3355, 3360, 3389,  
3505, 3597, 3619, 3781, 3819, 3955, 4042, 4044  
`\MT@xetex@no` ..... 244  
`\MT@xspace` ..... 2147, 2162  
`\MT@xspace@` ..... 2162  
`\MT@xunicodefalse` ..... 850  
`\MT@xunicodetrue` ..... 850, 851, 890  
  
**N** newcomputermodern (package) ..... 145  
newpx (package) ..... 21, 59, 145, 239  
newtx (package) ..... 21, 59, 145, 239  
  
**O** \outputmode ..... 194, 4117  
  
**P** \pdfadjustinterwordglue ..... 4390, 4684  
\pdfadjustspacing ..... 196, 948, 4349, 4353, 4645  
\pdfappendkern ..... 4392, 4717  
\pdfcopyfont ..... 1024  
\pdfcprot (package) ..... 26, 30, 54, 64, 99, 231  
\pdffeedback ..... 1053  
\pdffontexpand ..... 1537, 1540, 1544, 1550  
\pdffontsize ..... 1052, 1053  
\pdfnoligatures ..... 2283, 2299, 2305  
\pdfprependkern ..... 4391, 4716  
\pdfprotrudechars ..... 195, 4348, 4352, 4779  
pdfTeX (engine) ..... 1, 4, 6–9, 13–19,  
23–35, 37–39, 41–43, 45, 46, 53, 57–59, 64,  
70, 72–74, 76, 79, 80, 83, 85, 112, 115, 117,  
119, 129, 133–137, 139, 140, 197, 199, 231–237  
pdftexcmds (package) ..... 37, 237  
\pdftracingfonts ..... 358  
\pickup@font ..... 2905, 2907, 2916, 2939–2941  
pifont (package) ..... 99  
pinyin (package) ..... 56, 236, 237  
\protrudechars ..... 195, 4348  
protrusion (option) ..... 6, 11, 125, 126  
psnfss (package) ..... 79, 198, 239  
pstricks (package) ..... 134  
pxfonts (package) ..... 21, 145  
  
**Q** qfonts (package) ..... 21, 145, 233



<b>R</b>	ragged2e (package) . . . . .	26, 81, 138, 237, 238	\repeat . . . . .	3238
	reledmac (package) . . . . .	53, 239	\rightmarginkern . . . . .	822
	resize (package) . . . . .	108	\rpcode . . . . .	1221, 1262, 1263, 1462, 1465, 1865
	\remove@tlig . . . . .	2806		
<b>S</b>	selected (option) . . . . .	8, 14, 33, 69, 90, 126, 232	slantsc (package) . . . . .	177
	\selectfont . . . . .	1971, 3021, 3033, 4285, 4861	soul (package) . . . . .	4, 30, 56, 82, 235, 237, 240
	\set@fontsize . . . . .	3228	soulutf8 (package) . . . . .	30, 237
	\SetExpansion . . . . .	14, 36, 3468	\spacefactor . . . . .	2091
	\SetExtraKerning . . . . .	18, 38, 3526	\spaceskip . . . . .	2092, 2095
	\SetExtraSpacing . . . . .	19, 39, 3511	spacing (option) . . . . .	6, 11, 26, 31, 126, 237, 238
	\SetProtrusion . . . . .	13, 35, 3452	\stbscode . . . . .	1660, 1661, 1678, 1682, 1688
	\SetTracking . . . . .	15, 37, 3491	step (option) . . . . .	8, 31, 33, 128, 135, 137, 232, 234, 236
	\sfcode . . . . .	4701	step (package) . . . . .	21, 145, 240
	shapepar (package) . . . . .	29, 53	stix (package) . . . . .	21, 145, 240
	\shbscode . . . . .	1667, 1668, 1679, 1682, 1689	stix2 (package) . . . . .	21, 145, 240
	\showhyphens . . . . .	4631	stretch (option) . . . . .	8, 15, 25, 33, 128, 135, 137, 236
	shrink (option) . . . . .	8, 15, 33, 128, 135, 137, 234, 236	svjour3 (package) . . . . .	108, 240
<b>T</b>	\tagcode . . . . .	1860, 2286, 2292	tipa (package) . . . . .	99
	tempora (package) . . . . .	21, 145, 239	trace (package) . . . . .	101, 235
	tex4ht (package) . . . . .	54, 235	\tracingmicrotype . . . . .	87
	T <sub>E</sub> X Live (distribution) . . . . .	28, 39, 146, 233	\tracingmicrotypeinpdf . . . . .	98
	\textls . . . . .	23, 52, 53, 921, 945, 1880, 1966, 1983, 2188	\tracingmicrotypeinpdfallfalse . . . . .	105
	\textls* . . . . .	24	\tracingmicrotypeinpdfalltrue . . . . .	105
	\textmicrotypecontext . . . . .	22, 48, 917, 3025	tracking (option) . . . . .	6, 11, 23, 31, 126
	tikz (package) . . . . .	54, 227, 237, 239	txfonts (package) . . . . .	21, 145
	tikzposter (package) . . . . .	238		
<b>U</b>	ucs (package) . . . . .	95, 236	\UseMicrotypeSet . . . . .	12, 33, 60, 73, 3100, 3302
	ulgothic (package) . . . . .	21	\UTF@four@octets@noexpand . . . . .	2735
	\UnicodeEncodingName . . . . .	2816	\UTF@three@octets@noexpand . . . . .	2734
	unit (option) . . . . .	7, 32, 233, 235	\UTF@two@octets@noexpand . . . . .	2733
<b>V</b>	verbose (option) . . . . .	9, 31, 32, 127, 232, 233, 236		
<b>W</b>	wordcount (package) . . . . .	52, 236		
<b>X</b>	XCharter (package) . . . . .	21, 146, 239	\XeTeXglyphname . . . . .	2625
	xeCJK (package) . . . . .	27, 100, 238, 239	\XeTeXlastfontchar . . . . .	733
	X <sub>Y</sub> T <sub>E</sub> X (engine) . . . . .	1, 4, 6, 7, 9, 12, 13, 20, 29, 31, 34, 38, 40, 50, 53, 56, 59, 83, 93, 133, 137, 139, 144, 147, 151, 159, 238–240	\XeTeXprotrudechars . . . . .	4356
	\XeTeXcharglyph . . . . .	2619	\xspace . . . . .	2146, 2147
	\XeTeXfonttype . . . . .	2570	xspace (package) . . . . .	83, 238
	\XeTeXglyph . . . . .	1290	\xspaceskip . . . . .	2094, 2097
	\XeTeXglyphindex . . . . .	2606	xunicode (package) . . . . .	96, 97, 238
			xunicode-addon (package) . . . . .	100
<b>Y</b>	yfonts (package) . . . . .	24		
<b>Z</b>	zefonts (package) . . . . .	21, 144		

## E The L<sup>A</sup>T<sub>E</sub>X Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 L<sup>A</sup>T<sub>E</sub>X3 Project

### Preamble

The L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) is the primary license under which the L<sup>A</sup>T<sub>E</sub>X kernel and the base L<sup>A</sup>T<sub>E</sub>X packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

license may be particularly suitable if your work is T<sub>E</sub>X-related (such as a L<sup>A</sup>T<sub>E</sub>X package), but it is written in such a way that you can use it even if your work is unrelated to T<sub>E</sub>X.

The section ‘WHETHER AND HOW TO DISTRIBUTE

WORKS UNDER THIS LICENSE', below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the L<sup>A</sup>T<sub>E</sub>X3 Project, believe that the conditions below give you the freedom to make and distribute mod-

ified 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 'cfigguide.tex' and 'modguide.tex' in the base L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X-Format' or in the case of files belonging to the 'L<sup>A</sup>T<sub>E</sub>X-format' a program implementing the 'T<sub>E</sub>X language'.

## Conditions on Distribution and Modification

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
  - 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.
  - Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent refer-

- ence 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<sup>A</sup>T<sub>E</sub>X work, this could be done, for example, by posting to `comp.text.tex`.)
3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
  - (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
  4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention

is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

## Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

### Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘`modguide.tex`’ in the base L<sup>A</sup>T<sub>E</sub>X distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L<sup>A</sup>T<sub>E</sub>X under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L<sup>A</sup>T<sub>E</sub>X, 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<sup>A</sup>T<sub>E</sub>X-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.