

# The microtype package

Subliminal refinements towards typographical perfection

R Schlicht  
w.m.1@gmx.net

v2.7a  
2018/01/14

---

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

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

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 © 2004–2018 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 . . . . .	8
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>29</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 [51]	
14.2	Font setup . . . . .	56
	Protrusion [61] Expansion [68] Interword spacing (glue) [71] Additional kern- ing [72] Tracking [74] Disabling ligatures [84] Loading the configuration [86] Translating characters into slots [90] Hook into L <sup>A</sup> T <sub>E</sub> X's font selection [97] Context-sensitive setup [101]	
14.3	Configuration . . . . .	103
	Font sets [103] Variants and aliases [109] Disabling ligatures [110] Interaction with babel [110] Fine tuning [111] Character inheritance [117] Permuta- tion [119]	

14.4	Package options . . . . .	122
	Declaring the options [122] Loading the definition file [126] Reading the configuration file [127] Hook for other packages [128] Changing options later [128] Processing the options [131]	
<b>15</b>	<b>Configuration files</b>	<b>141</b>
15.1	Font sets . . . . .	141
15.2	Font variants and aliases . . . . .	142
15.3	Interaction with babel . . . . .	144
15.4	Note on admissible characters . . . . .	144
15.5	Character inheritance . . . . .	144
	OT1 [145] T1 [145] LY1 [146] OT4 [146] QX [147] T5 [147] EU1, EU2, TU [148] Euro symbols [149]	
15.6	Tracking . . . . .	149
15.7	Font expansion . . . . .	149
15.8	Character protrusion . . . . .	151
	Normal [152] Italics [160] Small caps [171] Italic small caps [174] Text companion [176] Computer Modern math [180] AMS symbols [184] Euler [188] Euro symbols [192]	
15.9	Interword spacing . . . . .	192
	Nonfrenchspacing [195]	
15.10	Additional kerning . . . . .	196
	French [196] Turkish [197]	
<b>16</b>	<b>OpenType configuration files</b>	<b>198</b>
16.1	Character inheritance . . . . .	198
16.2	Character protrusion . . . . .	202
<b>17</b>	<b>Auxiliary file for micro fine tuning</b>	<b>215</b>
<b>A</b>	<b>The title logo</b>	<b>217</b>
A.1	Macros . . . . .	217
A.2	Document . . . . .	222
<b>B</b>	<b>The letterspacing illustration</b>	<b>222</b>
B.1	Macros . . . . .	222
B.2	Document . . . . .	225
<b>C</b>	<b>Change history</b>	<b>227</b>
<b>D</b>	<b>Index</b>	<b>236</b>
<b>E</b>	<b>The L<sup>A</sup>T<sub>E</sub>X Project Public License</b>	<b>244</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 . . . . .	88

## 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 X<sub>Y</sub>T<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 `mimetype` 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 `mimetype` 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{mimetype}
```

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 `\mimicrotypesetup` 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  $\LaTeX$  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, preceded by an asterisk if it is contingent on the  $\TeX$  engine, version and/or the output 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  $\pdfTeX$  versions older than 1.20 or in DVI output mode (see section 3.5), or with  $X_{\text{Y}}\TeX$ . 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 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  $\pdfTeX$ ):

```
\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  $X_{\text{Y}}\TeX$  (you may use the `LetterSpace` option of the `fontspec` package instead). With  $\pdfTeX$ , 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  $X_{\text{Y}}\TeX$  or  $\text{Lua}\TeX$ .

Table 1:

Availability of micro-typographic features

T <sub>E</sub> X engine			Micro-typographic features					
Engine	Version	Output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
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>	∅	∅	☒ <sup>a</sup>
		PDF	★	★	★	∅	∅	☒
X <sub>Y</sub> T <sub>E</sub> X	≥ 0.9997	PDF	★	∅	∅	∅	∅	

★ = enabled    ☒ = not enabled    ∅ = not available                      <sup>a</sup> for legacy (TFM) fonts only

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 T<sub>E</sub>X engine with this capability and the output mode is PDF. If auto

is set to false, the font instances for all expansion steps must exist (with files called  $\langle font\ name \rangle \pm \langle expansion\ value \rangle$ , e.g., `cmr12+10`, as described in the [pdfTeX manual](#)).

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` fonts or use the Latin Modern fonts (package `lmodern`). With LuaTeX, expansion is always automatic, and also works in DVI mode (`dvilualatex`), however, because postprocessing programs like `dvips` or `dvipdfmx` are not (yet) capable of dealing with OpenType fonts, only for legacy fonts.

**stretch**  $\langle integer \rangle$  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**  $\langle integer \rangle$  \* 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>2</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**  $\langle integer \rangle$  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 1 em; admissible values are in the range of  $-1000$  to  $+1000$ .

### 3.5 Miscellaneous options

**DVIoutput** true, false \* false

pdfTeX and LuaTeX are not only able to generate PDF output but can also spit out DVI files.<sup>3</sup> The latter can be ordered with the option `DVIoutput`, which will set `\pdfoutput` to zero. For X<sub>Y</sub>TeX, this option is not applicable.

<sup>2</sup> 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.

<sup>3</sup> Recent TeX systems are using pdfTeX as the default engine even for DVI output.

Note that this will confuse packages that depend on the value of `\pdfoutput` if they were loaded earlier, as they had been made believe that they were called to generate PDF output where they actually weren't. These packages are, among others: `graphics`, `color`, `hyperref`, `pstricks` and, obviously, `ifpdf`. Either load these packages after `microtype` or else issue the command `\pdfoutput=0` earlier – in the latter case, the `DVIoutput` option is redundant.

When generating DVI files, font expansion has to be enabled explicitly. With `pdfTeX`, neither `letterspacing` nor *automatic* font expansion will work because the postprocessing drivers (`dvips`, `dvipdfm`, etc.) resp. the DVI viewer are not able to generate the fonts on the fly.

<code>draft</code>	true, false	false
<code>final</code>	If the <code>draft</code> option is passed to the package, <i>all micro-typographic extensions will be disabled</i> , which may lead to different line, and hence page, breaks. The <code>draft</code> and <code>final</code> 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 <code>draft</code> to show any overfull boxes, you should load <code>microtype</code> with the <code>final</code> option.	
<code>verbose</code>	true, false, errors, silent	false
	Information on the settings used for each font will be written into the log file if you enable the <code>verbose</code> option. When <code>microtype</code> 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 <code>verbose=errors</code> 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 <code>microtype</code> with <code>verbose=silent</code> .	
<code>babel</code>	true, false	false
	Loading the package with the <code>babel</code> option will adjust the typesetting according to the respective selected language. Read section 6 for further information.	
<code>config</code>	<i>(file name)</i>	<code>microtype</code>
	Various settings for this package will be loaded from a main configuration file, by default <code>microtype.cfg</code> (see section 5.7). You can have a different configuration file loaded instead by specifying its name <i>without the extension</i> , e.g., <code>config=mycrottype</code> .	

### 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 `incompatibility`, 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

<sup>4</sup> 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	∅	∅	∅	∅	∅
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	∅	∅	∅	∅
alltext-nott (allmath-nott)	Text encodings, TS1 (OML, OMS, U)	\rm*, \sf*	∅	∅	∅
basictext (basicmath)	Text encodings (OML, OMS)	\rm*, \sf*	\md*	∅	\normalsize, \footnotesize, \small, \large
smallcaps	Text encodings	∅	∅	\sc*,si,scit	∅
footnotesize	Text encodings, TS1	∅	∅	∅	-\small
scriptsize	Text encodings, TS1	∅	∅	∅	-\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', '1arge-').

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 used for character protrusion and additional kerning, the ‘basictext’ 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 “A 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\_1). 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 1 em 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 (`={\left},\right\}`), 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> Also note that pdfTeX prohibits the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the sloppy context has 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 58.

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 1 em (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 1 em.

**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 1 em (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 1 em (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 pdf $\TeX$ , this is not recommended, however, since it entails that kerning will be switched off, too. With Lua $\TeX$ , 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!

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.

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 50/1000em, 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.

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 in another way, this feature allows to modify 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, TeX 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 TeX 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, TeX has the concept of ‘space factors’. They may be used to increase the space after certain characters, most prominently the punctuation characters. pdfTeX’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  $\langle character \rangle = \langle spacing factors \rangle$ , 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  $\langle dimension \rangle$  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  $4$ ). 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 Lua<sub>TEX</sub> and Xe<sub>TEX</sub>, 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

Table 3:

Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
	Encodings [Scripts]	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
Adobe Garamond (pad, padx, padj)	OT1, T1, LY1, TS1	n, it, (sl) <sup>d</sup> , sc
URW Garamond (ugm) <sup>e</sup>	OT1, T1, TS1	n, it
Bitstream Letter Gothic (blg) <sup>f</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>g</sup>	OT1, OT4, T1, LY1, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Times (ptm, ptmx, ptmj) <sup>h</sup>	OT1, OT4, T1, LY1, QX, (TS1) <sup>a</sup>	n, it, (sl) <sup>d</sup> , sc
Latin Modern Roman	EU1/2, TU [Latin, Greek]	n, it, (sl) <sup>d</sup>
Charis SIL	EU1/2, TU [Latin, Cyrillic, Greek]	n, it, sc
Palatino Linotype <sup>i</sup>	EU1/2, TU [Latin]	n, it, 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

*a* Incomplete  
*b* Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)  
*c* Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr), XCharter  
*d* Settings inherited from italic shape  
*e* Aliases: mathdesign/URW Garamond (mdugm), garamondx (zgm, zgmj)  
*f* Alias: ulgothic (ulg)  
*g* Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T<sub>E</sub>X Gyre Pagella (qp1), newpx, FPL Neu (fp9x, fp9j)  
*h* Aliases: txfonts (txr), qfonts/QuasiTimes, T<sub>E</sub>X Gyre Termes (qtm), newtx, tempora  
*i* Aliases: T<sub>E</sub>X Gyre Pagella, Palatino LT Std, Palatino  
*j* Aliases: Latin Modern (lmsy, lmm)  
*k* Alias: eulervm (zeur, zeus)

fonts padx (expert set), padj (oldstyle numerals) and pad (plain) into one and the same file `mt-pad.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}
```

```
\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`  $\{ \langle font\ name \rangle \}$

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- $\langle font\ name \rangle$ .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`  $\{ \langle context\ assignments \rangle \}$

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}`  $\{ \langle context\ assignments \rangle \}$

`\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`  $\{ \langle context\ assignments \rangle \} \{ \langle general\ text \rangle \}$

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}
```

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
```

<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.

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 {<list of babel languages>} {<context list>}
```

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 [amount] {<general text>}
```

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

```
\lsstyle
```

in math mode – expects the text in the mandatory argument, while `\lsstyle` will 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.

<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.

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

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’ (‘ſt’) 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, ‘`Äu ṣi ḍt ṣl ọj i g̣f e i t`’, with ligatures shown in red, inhibited ligatures in green).

```
\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  $\LaTeX$ , 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`  $[ \langle \text{characters} \rangle ] \{ \langle \text{set of fonts} \rangle \}$

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}
```

*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`

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

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:

- If you want to use 8-bit characters in the configuration, you have to load the `inputenc` package first. Unicode input is also supported (when loading `inputenc` with the `utf8` or the `utf8x` option, or out of the box with `XYTeX` and `LuaTeX`). 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 LuaTeX, 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 pdfTeX, 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 pdfTeX in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your TeX system.
- ! pdfTeX error (font expansion): auto expansion is only possible with scalable fonts.  
Automatic font expansion has been improved in pdfTeX 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: pdflatex: 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 pdfTeX 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 TeX system is not set up to embed (or 'download') the base PostScript fonts (e.g., Times, Helvetica, Courier). In most TeX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to `true`.
- Warning: pdflatex (file *ecrm1000+20*): Font *ecrm1000+20 at 1200* not found  
Furthermore, pdfTeX 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 pdfTeX versions, this is only possible if you manually create expanded instances of the fonts.

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

- ! 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 (pdfTeX versions older than 1.30).  
When applying micro-typographic enhancement to a large document with a lot of fonts, pdfTeX 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., TeX Live, change the settings in `texmf.cnf`, for MiKTeX, in the file `miktex.ini` (2.4 or older) resp. `pdflatex.ini` (2.5 or newer).
- pdfTeX warning (font expansion): font should be expanded before its first use  
This warning will occur with pdfTeX 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 TeX 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 pdfTeX programme in the first place, which introduced the micro-typographic extensions and made them available to the TeX 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 X<sub>Y</sub>TeX 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 Gaille*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dybala*, *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’, ‘lcomdata’, *Reinhard Kotucha*, ‘jcr’, and *Nils Anders Danielsson*.

## 12 References

Hàn Thế Thành, ‘Micro-typographic extensions to the T<sub>E</sub>X 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 pdfT<sub>E</sub>X in practice’, in: *TUGBoat*, vol. 25 (2004), no. 1: ‘Proceedings of the Practical T<sub>E</sub>X 2004 Conference’, pp. 35–38. (Online at <http://www.tug.org/TUGboat/Articles/tb25-1/thanh.pdf>)

Hàn Thế Thành, ‘Font-specific issues in pdfT<sub>E</sub>X’, in: *TUGBoat*, vol. 29 (2008), no. 1: ‘EuroBachOT<sub>E</sub>X 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, *The pdfT<sub>E</sub>X user manual*, 20 April 2017. (Available from CTAN at </systems/doc/pdftex/manual/pdftex-a.pdf>)

<sup>13</sup> Available from CTAN at [pkg/ebgaramond](http://www.ctan.org/pkg/ebgaramond), including configuration files for microtype.

Karl Berry, *Fontname: Filenames for T<sub>E</sub>X fonts*, July 2009. (Available from CTAN at [/info/fontname/fontname.pdf](#))

L<sup>A</sup>T<sub>E</sub>X3 Project Team, *L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> font selection*, 27 November 2005. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](#))

Will Robertson, *The fontspec package: Font selection for X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and LuaL<sup>A</sup>T<sub>E</sub>X*, 9 September 2017. (Available from CTAN at [pkg/fontspec](#))

Élie Roux, Khaled Hosny, Philipp Gesang, *The luaotfload package*, 29 January 2017. (Available from CTAN at [pkg/luaotfload](#))

Carsten Schurig, Tobias Schlemmer, *The pdfcpot.sty package*, 10 June 2005. (Available from CTAN at [pkg/pdfcpot](#))

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at [pkg/soul](#)). See also Heiko Oberdiek's extension of this package, `soulutf8`, which adds Unicode support. (Available from CTAN at [pkg/soulutf8](#))

## 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.

### 2.7 (2017/07/07)

- Allow automatic expansion and letterspacing with LuaT<sub>E</sub>X in DVI mode (aka `dvilualatex`) [3.1, 3.3, table 1]
- Compatibility with L<sup>A</sup>T<sub>E</sub>X 2017/01/01 (fix warnings)

### 2.6 (2016/05/01)

- Support for LuaT<sub>E</sub>X  $\geq 0.85$
- Improvements for tracking/letterspacing with LuaT<sub>E</sub>X (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 LuaT<sub>E</sub>X and X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X
- Support for protrusion with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X  $\geq 0.9997$
- Support for tracking/letterspacing with LuaT<sub>E</sub>X  $\geq 0.62$
- Allow context-sensitive setup with LuaT<sub>E</sub>X
- 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 \sstyle 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':  $4 (\min(\text{stretch}, \text{shrink})/5)$  [3.3]
- Protrusion settings for Bitstream Charter

**1.4 (2004/11/12)**

- Set up fonts independently from  $\LaTeX$  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 `XYTeX` (`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`).
    - `pad`: Settings for Adobe Garamond (`mt-pad.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`).
    - `euroitc`: Settings for ITC Euro symbol fonts (`mt-euroitc.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
                  censo: it should be forbidden for packages to change catcodes within the preamble.

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

5 \let\MT@restore@catcodes\@empty
6 \def\MT@fix@catcode#1#2{%
7   \edef\MT@restore@catcodes{%
8     \MT@restore@catcodes
9     \catcode#1 \the\catcode#1\relax
10  }%
11  \catcode#1 #2\relax
12 }
13 (package)\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*\slig[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@n1#1{\MT@warning{#1\@gobble}}
78 < *package >
\MT@info@nl 79 \def\MT@info{\PackageInfo\MT@MT}
\MT@vinfo 80 \def\MT@info@n1#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@n1 1: + sets & lists
2: + heirs
3: + slots
4: + factors
```

```
87 < *debug >
88 \MT@warning@n1{This is the debug version}
89 \newcount\tracingmicrotype
```

```

90 \tracingmicrotype=2
91 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
92 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1@gobble}\MT@addto@annot{#1}}
93 \let\MT@vinfo\MT@info@n1
94 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
95 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1@gobble}\MT@addto@annot{Warning: #1}}
96 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
97 \def\MT@dinfo@n1#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@n1{#2}\fi}

```

\tracingmicrotypeinpdf

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

Cases for \tracingmicrotypeinpdf:

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

```
98 \newcount\tracingmicrotypeinpdf
```

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

```
\tracingmicrotypeinpdf=2
```

```

\MT@pdf@annot
\MT@addto@annot
\ifMT@inannot

```

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

```

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

```

\iftracingmicrotypeinpdfall

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

```
105 \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdfannot

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

```

106 \def\MT@show@pdfannot#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\hmode@bgroup{\leavevmode\bgroup}
130   \def\nfss@text#1{\mbox{#1}}
131   \let\@typeset@protect\relax
132   \ifx\epain@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 <debug>\MT@info@n1{0}{this is
159 <debug>^^Q not
160 <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.

```

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

```

A hack circumventing the T<sub>E</sub>X Live 2004 hack which undefines the pdf<sub>T</sub>E<sub>X</sub> primitives in the format in order to hide the fact that pdf<sub>T</sub>E<sub>X</sub> is being run from the

user. This has been *fixed* in T<sub>E</sub>X Live 2005.

```
168 \ifx\normalpdftexversion\@undefined \else
169   \let\pdftexversion \normalpdftexversion
170   \let\pdftexrevision\normalpdftexrevision
171   \let\pdfoutput      \normalpdfoutput
172 \fi
```

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

```
\MT@engine@toold 173 \let\MT@engine\relax
174 <letterspace>\def\MT@engine@toold{0}
175 \ifx\pdftexversion\@undefined \else
176   \ifx\pdftexversion\relax \else
177     \def\MT@engine{pdf}
178   <letterspace> \let\MT@pdf@or@lua\@firstoftwo
179 <letterspace> \ifnum\pdftexversion > 139 \def\MT@engine@toold{1}\fi
180 \fi
181 \fi
182 \ifx\directlua\@undefined \else
183   \ifx\directlua\relax \else
184     \def\MT@engine{lua}
```

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

```
185 <*letterspace>
186   \let\MT@pdf@or@lua\@secondoftwo
187   \ifnum\luatexversion < 62 \def\MT@engine@toold{0}
188   \else
189     \def\MT@engine@toold{1}
190     \ifnum\luatexversion > 84
191       \let\pdfoutput\outputmode
192       \let\pdfprotrudechars\protrudechars
193     \fi
194   \fi
195 </letterspace>
196 \fi
197 \fi
198 <*package>
199 \ifx\MT@engine\relax
200   \ifx\XeTeXversion\@undefined \else
201     \ifx\XeTeXversion\relax \else
202       \def\MT@engine{xe}
203     \fi
204   \fi
205 \fi
206 </package>
207 </package|letterspace>
```

`\MT@pdftex@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@pdftex@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 1 em ( $\geq$  0.14h)
- 4: + automatic font expansion; protrusion no longer has to be set up first; scale factor fixed to 1000; default `\efcode` = 1000 ( $\geq$  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` ( $\geq 1.40.4$ )

```

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

```

`\MT@xetex@no`    X<sub>Y</sub>TeX supports character protrusion since version 0.9997.

```

240 (*xetex-def)
241 (debug)\MT@dinfol{0}{this is xetex (\the\XeTeXversion\XeTeXrevision)}
242 \ifdim 0\XeTeXrevision pt < 0.9997pt
243   \def\MT@xetex@no{1}
244 \else
245   \def\MT@xetex@no{2}
246 \fi
247 (debug)\MT@dinfol{0}{xetex no.: \MT@xetex@no}
248 (/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 ( $\geq 0.36$ )
- 3: + `\letterspacefont` ( $\geq 0.62$ )
- 4: + almost all of the pdfTeX primitives have been renamed ( $\geq 0.85$ )

---

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

5: + default \efcode = 1000; \protrusionboundary [not yet supported] ( $\geq 0.90$ )

```
249 (*luatex-def)
250 (debug)\MT@info@n10{this is luatex (\the\luatexversion)}
```

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

```
251 \def\MT@lua{\directlua}
252 \def\MT@luatex@no{5}
253 \ifnum\luatexversion<90
254   \def\MT@luatex@no{4}
255   \ifnum\luatexversion<85
256     \def\MT@luatex@no{3}
257     \ifnum\luatexversion<62
258       \def\MT@luatex@no{2}
259       \ifnum\luatexversion<36
260         \def\MT@lua{\directlua0}
261         \def\MT@luatex@no{1}
262       \fi
263     \fi
264   \fi
265 \fi

266 (debug)\MT@info@n10{luatex no.: \MT@luatex@no}
267 (/luatex-def)

268 (*pdftex-def|xetex-def|letterspace)
269 \ifnum
270 (pdftex-def|xetex-def) \csname MT@MT@engine tex@no\endcsname < 2
271 (letterspace) \MT@engine@toold=\z@
272 \MT@warning@n1{You
273 (*letterspace)
274   \ifx\MT@engine\relax
275     don't seem to be using pdftex or luatex.\MessageBreak
276     Try running `pdftex' or `luatex' instead of.\MessageBreak
277     `\ifx\XeTeXversion\@undefined\else xel\fi tex'%
278   \else
279 (/letterspace)
280     are using a \MT@engine tex version older than
281 (pdftex-def) 0.14f%
282 (xetex-def) 0.9997%
283 (letterspace) \MT@pdf@or@lua{1.40}{0.62}%
284   .\MessageBreak
285   `\MT@MT' does not work with this version.\MessageBreak
286   Please install a newer version of \MT@engine tex%
287 (letterspace) \fi
288   .\MessageBreak I will quit now}
289 \MT@clear@options
290 \endinput\fi
291 (/pdftex-def|xetex-def|letterspace)
```

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

```
292 (*package|letterspace)
293 \RequirePackage{keyval}[1997/11/10]
294 (*package)
```

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

```
295 \newtoks\MT@toks
```

`\ifMT@if@` A scratch if.

```
296 \newif\ifMT@if@
```

### 14.1.3 Declarations

```

\ifMT@protrusion    These are the global switches ...
\ifMT@expansion    297 \newif\ifMT@protrusion
  \ifMT@auto        298 \newif\ifMT@expansion
  \ifMT@selected    299 \newif\ifMT@auto
\ifMT@noligatures  300 \newif\ifMT@selected
  \ifMT@draft        301 \newif\ifMT@noligatures
  \ifMT@spacing      302 \newif\ifMT@draft
  \ifMT@kerning      303 \newif\ifMT@spacing
  \ifMT@tracking     304 \newif\ifMT@kerning
  \ifMT@babel        305 \newif\ifMT@tracking
  \ifMT@babel        306 \newif\ifMT@babel
  \ifMT@babel        ... and numbers.
  \MT@ex@level       307 \let\MT@pr@level\tw@
  \MT@pr@factor      308 \let\MT@ex@level\tw@
  \MT@ex@factor      309 \let\MT@pr@factor\@m
  \MT@sp@factor      310 \let\MT@ex@factor\@m
  \MT@kn@factor      311 \let\MT@sp@factor\@m
  \MT@kn@factor      312 \let\MT@kn@factor\@m
  \MT@pr@unit        Default unit for protrusion settings is character width, for spacing space, for kerning
  \MT@sp@unit        (and tracking) 1 em.
  \MT@kn@unit        313 \let\MT@pr@unit\@empty
  \MT@kn@unit        314 \let\MT@sp@unit\m@ne
  \MT@kn@unit        315 \def\MT@kn@unit{1em}

  \MT@stretch        Expansion settings.
  \MT@shrink         316 \let\MT@stretch\m@ne
  \MT@step           317 \let\MT@shrink \m@ne
  \MT@step           318 \let\MT@step \m@ne

  \MT@pr@min         Minimum and maximum values allowed by pdfTeX.
  \MT@pr@max         319 \def\MT@pr@min{-\@m}
  \MT@ex@min         320 \let\MT@pr@max\@m
  \MT@ex@min         321 \let\MT@ex@min\z@
  \MT@ex@max         322 \let\MT@ex@max\@m
  \MT@sp@min         323 \def\MT@sp@min{-\@m}
  \MT@sp@max         324 \let\MT@sp@max\@m
  \MT@kn@min         325 \def\MT@kn@min{-\@m}
  \MT@kn@max         326 \let\MT@kn@max\@m
  \MT@kn@max         327 /package
  \MT@tr@min         328 \def\MT@tr@min{-\@m}
  \MT@tr@max         329 \let\MT@tr@max\@m
  \MT@tr@max         330 *package

\MT@factor@default  Default factor.
  \MT@factor@default 331 \def\MT@factor@default{1000 }

\MT@stretch@default  Default values for expansion.
\MT@shrink@default  332 \def\MT@stretch@default{20 }
  \MT@shrink@default 333 \def\MT@shrink@default{20 }

  \MT@letterspace    Default value for letterspacing (in thousandths of 1 em).
\MT@letterspace@default 334 /package
  \MT@letterspace@default 335 \let\MT@letterspace\m@ne
  \MT@letterspace@default 336 \def\MT@letterspace@default{100}
  \MT@letterspace@default 337 *package

\ifMT@document      Our private test whether we're still in the preamble.
  \ifMT@document    338 \newif\ifMT@document
  \ifMT@document    339 /package
  \ifMT@document    340 /package|letterspace

```

### 14.1.4 Auxiliary macros

`\MT@requires@pdftex` For definitions that depend on a particular pdf $\TeX$  resp. Lua $\TeX$  version.

```

\MT@requires@luatex 341 <*pdftex-def|luatex-def>
342 \def
343 <pdftex-def> \MT@requires@pdftex%
344 <luatex-def> \MT@requires@luatex%
345 #1{\ifnum
346 <pdftex-def> \MT@pdftex@no
347 <luatex-def> \MT@luatex@no
348 <#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi>
349 <luatex-def&debug>\MT@requires@luatex4{\directlua{tex.enableprimitives('pdf',{'tracingfonts'})}}\relax
350 <pdftex-def&debug>\MT@requires@pdftex6{
351 <debug>\pdftracingfonts=1
352 <pdftex-def&debug>}\relax
353 </pdftex-def|luatex-def>

```

Some functions are loaded from a dedicated lua file. This avoids character escaping problems and incompatibilities between versions of Lua $\TeX$ . Unless running a recent L $\TeX$ , we load the `luatexbase` package.

```

354 <*luatex-def>
355 \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 Lua $\TeX$  documents that don't load this package, anyway.

```

356 \RequirePackage{luaotfload}
357 \MT@lua{require("microtype")}
358 </luatex-def>

```

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

```

359 <*luafile>
360
361 local err, warn, info, log = luatexbase.provides_module(microtype.module)
362 microtype.warning = warn
363
364 local find      = string.find
365 local match    = string.match
366 local tex_write = tex.write
367
368 local catpackage
369 if luatexbase.registernumber then
370   catpackage = luatexbase.registernumber("catcodetable@atletter") -- LaTeX
371 else
372   catpackage = luatexbase.catcodetables.CatcodeTableAtletter -- luatexbase
373 end
374 function microtype.sprint (...)
375   tex.sprint(catpackage, ...)
376 end
377
378 </luafile>

```

To be continued, but first back to primitives.

`\MT@glet` Here's the forgotten one.

```

379 <*package|letterspace>
380 \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 381 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
382 <*package>
383 \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 384 \def\MT@def@n{\MT@exp@cs\def}
385 \def\MT@gdef@n{\MT@exp@gcs\gdef}

\MT@edef@n      Its expanding versions.
\MT@xdef@n 386 </package>
387 \def\MT@edef@n{\MT@exp@cs\edef}
388 <*package>
389 \def\MT@xdef@n{\MT@exp@gcs\xdef}

\MT@let@nc      \let a \csname sequence to a command.
\MT@glet@nc 390 \def\MT@let@nc{\MT@exp@cs\let}
391 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

\MT@let@cn      \let a command to a \csname sequence.
392 </package>
393 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
394 <*package>

\MT@let@nn      \let a \csname sequence to a \csname sequence.
\MT@glet@nn 395 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
396 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

\MT@font      Remove trailing space from the font name.
397 \def\MT@font{\expandafter\string\MT@font}

\MT@exp@one@n      Expand the second token once and enclose it in braces.
398 </package>
399 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}

\MT@exp@two@c      Expand the next two tokens after <#1> once.
400 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
401 <*package>

\MT@exp@two@n      Expand the next two tokens after <#1> once and enclose them in braces.
402 \def\MT@exp@two@n#1#2#3{%
403   \expandafter\expandafter\expandafter
404   #1\expandafter\expandafter\expandafter
405   {\expandafter#2\expandafter}\expandafter{#3}}

You do not wonder why \MT@exp@one@c doesn't exist, do you?
\MT@ifdefined@c@T      Wrapper for testing whether command resp. \csname sequence is defined. If we
\MT@ifdefined@c@TF      are running e-TeX, we will use its primitives \ifdefined and \ifcsname, which
\MT@ifdefined@n@T      decreases memory use substantially.
\MT@ifdefined@n@TF 406 \def\MT@ifdefined@c@T#1{%
407   ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
408   ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
409 }
410 </package>
411 \def\MT@ifdefined@c@TF#1{%
412   ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
413   <package>^^Q \ifx#1\@undefined
414   <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
415 }
416 \def\MT@ifdefined@n@T#1{%
417   ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
418   <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
419   <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
420 }
421 \def\MT@ifdefined@n@TF#1{%
422   ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
423   <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
424   <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi

```

```

425 }
426 <*/package>

\MT@detokenize@n    Translate a macro into a token list. With e-TeX, we can use \detokenize. We also
\MT@detokenize@c    need to remove the last trailing space; and only the last one – therefore the fiddling
\MT@rem@last@space  (and the \string isn't perfect, of course).
427 \def\MT@detokenize@n#1{%
428 ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil
429 ^^Q \string#1%
430 }
431 \def\MT@detokenize@c#1{%
432 ^^X \MT@exp@one@n\MT@detokenize@n#1%
433 ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
434 }
435 \def\MT@rem@last@space#1 #2{#1%
436 \ifx\@nil#2\else \space
437 \expandafter\MT@rem@last@space\expandafter#2\fi
438 }

\MT@ifempty    Test whether argument is empty.
439 </package>
440 \begingroup
441 \catcode`\%=12
442 \catcode`\&=14
443 \gdef\MT@ifempty#1{&
444 \if %#1&
445 \expandafter\@firstoftwo
446 \else
447 \expandafter\@secondoftwo
448 \fi
449 }
450 \endgroup
451 <*/package>

\MT@ifint    Test whether argument is an integer, using an old trick by Mr. Arseneau, or the
              latest and greatest from pdfTeX or LuaTeX (which also allows negative numbers, as
              required by the letterspace option).
452 </package>
453 </package|letterspace>
454 <pdfTeX-def>\MT@requires@pdftex6{
455 <letterspace>\MT@pdf@or@lua{
456 <*/pdfTeX-def|letterspace>
457 \def\MT@ifint#1{%
458 \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
459 \expandafter\@secondoftwo
460 \else
461 \expandafter\@firstoftwo
462 \fi
463 }
464 }{
465 </pdfTeX-def|letterspace>
466 <*/pdfTeX-def|xetex-def|letterspace>
467 \def\MT@ifint#1{%
468 \if!\ifnum9<1#1!\else?\fi
469 \expandafter\@firstoftwo
470 \else
471 \expandafter\@secondoftwo
472 \fi
473 }
474 </pdfTeX-def|xetex-def|letterspace>
475 <pdfTeX-def|letterspace>}
476 <luaTeX-def>\def\MT@ifint#1{\csname\MT@lua{microtype.if_int}([\#1])\endcsname}
477 <*/luafile>
478 local function if_int(s)

```

```

479 if find(s,"^-[0-9]+ *$") then
480   tex_write("@firstoftwo")
481 else
482   tex_write("@secondoftwo")
483 end
484 end
485 microtype.if_int = if_int
486
487 /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.)

```

488 *pdfTeX-def
489 \MT@requires@pdfTeX6{
490 \def\MT@ifdimen#1{%
491   \ifcase\pdfmatch{^[0-9]+([.,][0-9]+)?[.,][0-9]+}%
492     (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
493   \expandafter\@secondoftwo
494   \else
495     \expandafter\@firstoftwo
496   \fi
497 }
498 }{
499 /pdfTeX-def
500 *pdfTeX-def|xetex-def
501 \def\MT@ifdimen#1{%
502   \setbox\z@=\hbox{%
503     \MT@count=1#1\relax
504     \ifnum\MT@count=\@ne
505       \aftergroup\@secondoftwo
506     \else
507       \aftergroup\@firstoftwo
508     \fi
509   }%
510 }
511 /pdfTeX-def|xetex-def
512 pdfTeX-def}
513 luaTeX-def\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen}[[#1]]\endcsname}
514 *luafile
515 local function if_dimen(s)
516   if (find(s, "^-[0-9]+(%a*) *$") or
517       find(s, "^-[0-9]*[.,][0-9]+(%a*) *$")) then
518     tex_write("@firstoftwo")
519   else
520     tex_write("@secondoftwo")
521   end
522 end
523 microtype.if_dimen = if_dimen
524
525 /luafile

```

`\MT@ifdim` Test floating point numbers.

```

526 *package
527 \def\MT@ifdim#1#2#3{%
528   \ifdim #1\p@ #2 #3\p@
529     \expandafter\@firstoftwo
530   \else
531     \expandafter\@secondoftwo
532   \fi
533 }
534 /package

```

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

```

535 *pdfTeX-def
536 \MT@requires@pdfTeX5{

```

```

537 \def\MT@ifstreq#1#2{%
538   \ifcase\pdfstrcmp{#1}{#2}\relax
539   \expandafter\@firstoftwo
540   \else
541   \expandafter\@secondoftwo
542   \fi
543 }
544 }{
545 /pdftex-def
546 (*pdftex-def|xetex-def
547 \def\MT@ifstreq#1#2{%
548   \edef\MT@res@a{#1}%
549   \edef\MT@res@b{#2}%
550   \ifx\MT@res@a\MT@res@b
551   \expandafter\@firstoftwo
552   \else
553   \expandafter\@secondoftwo
554   \fi
555 }
556 /pdftex-def|xetex-def
557 pdftex-def}
558 luatex-def\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq}([[#1]],[[#2]])\endcsname}
559 *luafile
560 local function if_str_eq(s1, s2)
561   if s1 == s2 then
562     tex_write("@firstoftwo")
563   else
564     tex_write("@secondoftwo")
565   end
566 end
567 microtype.if_str_eq = if_str_eq
568
569 /luafile

```

`\MT@xadd`     Add item to a list.

```

570 (*package
571 \def\MT@xadd#1#2{%
572   \ifx#1\relax
573   \xdef#1{#2}%
574   \else
575   \xdef#1{#1#2}%
576   \fi
577 }

```

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

```

578 \def\MT@xaddb#1#2{%
579   \ifx#1\relax
580   \xdef#1{#2}%
581   \else
582   \xdef#1{#2#1}%
583   \fi
584 }
585 /package

```

`\MT@map@clist@n`     Run `<#2>` on all elements of the comma list `<#1>`. This and the following is modelled after L<sup>A</sup>T<sub>E</sub>X3 commands.

`\MT@map@clist@c`

`\MT@map@clist@t`     *(\*package|letterspace*

`\MT@clist@function`     `\def\MT@map@clist@n#1#2{%`

```

588   \ifx\@empty#1\else
589   \def\MT@clist@function##1{#2}%
590   \MT@map@clist@#1,\@nil,\@nnil
591   \fi
592 }

```

```

593 \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}

```

```

594 \def\MT@map@clist@#1,{%
595   \ifx\@nil#1%
596     \expandafter\MT@clist@break
597   \fi
598   \MT@clist@function{#1}%
599   \MT@map@clist@
600 }
601 \let\MT@clist@function@gobble
602 \def\MT@clist@break#1\@nnil{}
603 (*package)

```

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

```

\MT@map@tlist@c
\MT@map@tlist@ 604 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break 605 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
606 \def\MT@map@tlist@#1#2{%
607   \ifx\@nnil#2\else
608     #1{#2}%
609     \expandafter\MT@map@tlist@
610     \expandafter#1%
611   \fi
612 }
613 \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 614 \newif\ifMT@inlist@
615 \def\MT@in@clist#1#2{%
616   \def\MT@res@a#1,#1,##2##3\@nnil{%
617     \ifx##2\empty
618       \MT@inlist@false
619     \else
620       \MT@inlist@true
621     \fi
622   }%
623   \expandafter\MT@res@a\expandafter,#2,#1,\empty\@nnil
624 }

```

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

```

625 \def\MT@rem@from@clist#1#2{%
626   \def\MT@res@a#1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
627   \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\empty\else##1\fi}%
628   \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
629 }

```

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

```

\MT@in@tlist@ 630 \def\MT@in@tlist#1#2{%
631   \MT@inlist@false
632   \def\MT@res@a{#1}%
633   \MT@map@tlist@c#2\MT@in@tlist@
634 }
635 \def\MT@in@tlist@#1{%
636   \edef\MT@res@b{#1}%
637   \ifx\MT@res@a\MT@res@b
638     \MT@inlist@true
639     \expandafter\MT@tlist@break
640   \fi
641 }

```

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

```

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

```

```

642 \def\MT@in@rlist#1{%
643   \MT@inlist@false
644   \MT@map@tlist@c#1\MT@in@rlist@
645 }
646 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@#1}
647 \def\MT@in@rlist@#1#2#3{%
648   \MT@ifdim{#2}=\m@ne{%
649     \MT@ifdim{#1}=\MT@size
650     \MT@inlist@true
651     \relax
652   }%
653   \MT@ifdim\MT@size<{#1}\relax{%
654     \MT@ifdim\MT@size<{#2}%
655     \MT@inlist@true
656     \relax
657   }%
658 }%
659 \ifMT@inlist@
660   \def\MT@size@name{#3}%
661   \expandafter\MT@tlist@break
662 \fi
663 }

```

`\MT@loop` This is the same as L<sup>A</sup>T<sub>E</sub>X's `\loop`, which we mustn't use, since this could confuse an outer `\loop` in the document.

`\MT@iterate`

```

\MT@repeat 664 </package>
665 \def\MT@loop#1\MT@repeat{%
666   \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
667   \MT@iterate \let\MT@iterate\relax
668 }
669 \let\MT@repeat\fi

```

`\MT@while@num` Execute `<#3>` from `<#1>` up to (excluding) `<#2>` (much faster than L<sup>A</sup>T<sub>E</sub>X's `\@whilenum`).

```

670 \def\MT@while@num#1#2#3{%
671   \@tempcnta#1\relax
672   \MT@loop #3%
673   \advance\@tempcnta \@ne
674   \ifnum\@tempcnta < #2\MT@repeat
675 }
676 </package|letterspace>

```

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

```

677 <pdfTeX-def|letterspace>\def\MT@do@font{\MT@while@num\z@\cclvi}

```

resp. for the whole font for LuaT<sub>E</sub>X, if loaded by `fontspec/luatfload`.

```

678 <*luatex-def>
679 \def\MT@do@font#1{%
680   \MT@if@fontspec@font{%
681     \def\MT@do@font@function{#1}%
682     \MT@lua{microtype.do_font()}%
683   }{\MT@while@num\z@\cclvi{#1}}%
684 }
685 </luatex-def>

```

This is the lua function, which is much faster than looping through all glyphs in T<sub>E</sub>X. Legacy fonts (which this function might be fed with, because `fontspec` isn't always getting it right) don't contain a `v.index` field.

```

686 <*luafile>
687 local function do_font()
688   if fonts then
689     local thefont
690     if fonts.ids then --- legacy luatfload
691       thefont = fonts.ids[font.current()]
692     else --- new location

```

```

693     thefont = fonts.hashes.identifiers[font.current()]
694   end
695   if thefont then
696     for i,v in next,thefont.characters do
697       if v.index == nil or v.index > 0 then
698         microtype.sprint([[ \@tempcnta=]]..i..[[\relax\MT@dofont@function]])
699       end
700     end
701   end
702 end
703 end
704 microtype.do_font = do_font
705
706 /luafile

```

The X<sub>Y</sub>TeX variant.

```

707 (*xetex-def)
708 \def\MT@do@font#1{%
709   \@tempcnta=\z@
710   \MT@loop #1%
711   \advance\@tempcnta \@ne
712   \ifnum\@tempcnta < \XeTeXcountglyphs\MT@font \MT@repeat
713 }
714 /xetex-def
715 (*package)

```

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

`\MT@increment`

```

716 \newcount\MT@count
717 \def\MT@increment#1{%
718   ^^X \edef#1{\number\numexpr #1 + 1\relax}%
719   ^^Q \MT@count=#1\relax
720   ^^Q \advance\MT@count \@ne
721   ^^Q \edef#1{\number\MT@count}%
722 }

```

`\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.

```

723 \def\MT@scale#1#2#3{%
724   ^^Q \multiply #1 #2\relax
725   \ifnum #3 = \z@
726     ^^X #1=\numexpr #1 * #2\relax
727   \else
728     ^^X #1=\numexpr #1 * #2 / #3\relax
729   ^^Q \divide #1 #3\relax
730   \fi
731 }

```

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

`\MT@abbr@ex`

```

732 \def\MT@abbr@pr{protrusion}
733 \def\MT@abbr@ex{expansion}
734 \def\MT@abbr@pr@c{protrusion codes}
735 \def\MT@abbr@ex@c{expansion codes}
736 \def\MT@abbr@pr@inh{protrusion inheritance}
737 \def\MT@abbr@ex@inh{expansion inheritance}
738 \def\MT@abbr@n1{no ligatures}
739 \def\MT@abbr@sp{spacing}
740 \def\MT@abbr@sp@c{interword spacing codes}
741 \def\MT@abbr@sp@inh{interword spacing inheritance}
742 \def\MT@abbr@kn{kerning}

```

`\MT@abbr@pr@c`

`\MT@abbr@ex@c`

`\MT@abbr@pr@inh`

`\MT@abbr@ex@inh`

`\MT@abbr@n1`

`\MT@abbr@sp`

`\MT@abbr@sp@c`

`\MT@abbr@sp@inh`

`\MT@abbr@kn`

`\MT@abbr@kn@c`

`\MT@abbr@kn@inh`

`\MT@abbr@tr`

`\MT@abbr@tr@c`

```

743 \def\MT@abbr@kn@c{kerneling codes}
744 \def\MT@abbr@kn@inh{kerneling inheritance}
745 \def\MT@abbr@tr{tracking}
746 \def\MT@abbr@tr@c{tracking amount}

\MT@rba@protrusion    These we also need the other way round.
\MT@rba@expansion    747 \def\MT@rba@protrusion{pr}
\MT@rba@spacing      748 \def\MT@rba@expansion{ex}
\MT@rba@kerning      749 \def\MT@rba@spacing{sp}
\MT@rba@tracking      750 \def\MT@rba@kerning{kn}
\MT@rba@tracking      751 \def\MT@rba@tracking{tr}

\MT@features          We can work on these lists to save some guards in the dtx file.
\MT@features@long    752 \def\MT@features{pr,ex,sp,kn,tr}
                    753 \def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}

\MT@is@feature        Whenever an optional argument accepts a list of features, we can use this com-
                    mand to check whether a feature exists in order to prevent a rather confusing
                    ‘Missing \endcsname inserted’ error message. The feature (long form) must be in
                    <#1>, the type of list to ignore in <#2>, then comes the action.
                    754 \def\MT@is@feature#1#2{%
                    755   \MT@in@clist{#1}\MT@features@long
                    756   \ifMT@in@list@
                    757     \expandafter\@firstofone
                    758   \else
                    759     \MT@error{`#1' is not an available micro-typographic\MessageBreak
                    760       feature. Ignoring #2}{Available features are: `~\MT@features@long'.}%
                    761     \expandafter\@gobble
                    762   \fi
                    763 }

```

### 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.

```

764 \@ifl@aded{tex}{wordcount}{%
765   \MT@warning@nl{Detected the `wordcount' utility.\MessageBreak
766     Disabling `~\MT@MT', since it wouldn't work}%
767   \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.

```

768 \@ifclassloaded{minimal}{%
769   \MT@warning@nl{Detected the `minimal' class.\MessageBreak
770     Expect lots of warnings and some malfunctions.\MessageBreak
771     You might want to use a proper class instead}%
772 }\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.

```

773 </package>
774 <*package|letterspace>
775 <plain>\MT@requires@latexl{
776 \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.
777 \def\MT@addto@setup{\g@addto@macro\MT@setup@

                   Don't hesitate with miniltx.
778 <plain>}{\let\MT@addto@setup@firstofone}

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

\MT@with@babel@and@T    LATEX's \@ifpackagewith ignores the class options.
782 \def\MT@with@babel@and@T#1{%
783   \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
784     \expandtwoargs\MT@in@clist{#1}
785     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
786     \ifMT@inlist@expandafter@gobble\fi
787   }@gobble
788 }

\MT@ledmac@setup    The ledmac package first saves each paragraph in a box, from which it then splits
                   off the lines one by one. This will destroy character protrusion. (There aren't any
                   problems with the lineno package, since it takes a different approach.) — ... —
                   After much to and fro, the situation has finally settled and there is a fix. Beginning
                   with pdfTEX version 1.21b together with ledpatch.sty as of 2005/06/02 (v0.4),
                   character protrusion will work at last.
                   Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to
                   allow for protrusion. \leftmarginkern and \rightmarginkern are new primitives
                   of pdfTEX 1.21b (aka. 1.30.0). They are also part of recent XYTEX. The successor
                   packages eledmac and reledmac are also supported.
789 </package>
790 <pdftex-def>\MT@requires@pdftex5{
791 <*pdftex-def|luatex-def|xetex-def>
792   \def\MT@ledmac@setup{%
793     \ifMT@protrusion
794       \MT@ifdefined@c@TF\l@dunhbox@line{%

\MT@led@unhbox@line    Hook.
795     \MT@info@nl{Patching ((r)e)ledmac to enable character protrusion}%
796     \let\MT@led@unhbox@line\l@dunhbox@line
797     \renewcommand*{\l@dunhbox@line}[1]{%
798       \ifhbox##1%
799         \kern\leftmarginkern##1%
800         \expandafter\MT@led@unhbox@line\expandafter##1\expandafter
801         \kern\righmarginkern##1%
802       \fi
803     }%
804   }{%
805     \MT@warning@nl{%
806       Character protrusion in paragraphs with line\MessageBreak
807       numbering will only work if you update ledmac,\MessageBreak
808       or use one of its successors, eledmac or reledmac}%
809   }%
810   \fi
811 }

```

```

812 </pdfTeX-def|luatex-def|xetex-def>
813 <{*pdfTeX-def}>
814 }{
815   \def\MT@ledmac@setup{%
816     \ifMT@protrusion
817       \MT@warning@n1{%
818         The pdfTeX version you are using does not allow\MessageBreak
819         character protrusion in paragraphs with line\MessageBreak
820         numbering by the `((r)e)ledmac' package.\MessageBreak
821         Upgrade pdfTeX to version 1.30 or later}%
822     \fi
823   }
824 }
825 </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 `\#`.

```

826 <{*package|letterspace}>
827 <{*package}>
828 \def\MT@restore@p@h{\chardef\%`%\ \chardef\#`#\# }

```

`\ifMT@unicode` Two new conditionals for use with X<sub>Y</sub>T<sub>E</sub>X or Lua<sub>T</sub>E<sub>X</sub>.

```

\ifMT@fontspec 829 \newif\ifMT@unicode
830 \MT@with@package@T{xunicode}\MT@xunicodetrue
831 </package>
832 \newif\ifMT@fontspec
833 <letterspace>\MT@requires@lateX2{
834 \MT@with@package@T{fontspec}\MT@fontspectrue
835 <letterspace>}\MT@fontspecfalse}

```

`\MT@if@fontspec@font` For fonts loaded by fontspec (or, rather, luaotfload) we can use some of the features the latter package provides.

`\MT@fontspec@setup`

```

836 \let\MT@if@fontspec@font\@secondoftwo
837 \def\MT@fontspec@setup{%
838   \ifpackagelater{fontspec}{2013/05/23}{
839     \MT@let@cn\MT@if@fontspec@font{fontspec_if_fontspec_font:TF}%
840   }\relax
841 }
842 \ifMT@fontspec\MT@fontspec@setup\fi

```

`\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`

```

843 <{*package}>
844 \let\MT@maybe@gobble@with@tikz\@firstofone
845 \def\MT@tikz@setup{%
846   \def\MT@maybe@gobble@with@tikz{%
847     \ifnum\tikz@expandcount>\z@
848       \expandafter\@gobble
849     \else
850       \expandafter\@firstofone
851     \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`.)

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

When a font is defined via `\fontspec`, the font is not actually loaded, hence  $\text{\XeTeX}$  resp.  $\text{\LuaTeX}$  would see a wrong font (in `\MT@get@slot`). Therefore, we load the current font.

```
853 \ifMT@fontspec\MT@font\fi
```

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

```
854 \MT@if@false
855 \MT@with@babel@and@T{spanish} \MT@if@true
856 \MT@with@babel@and@T{galician} \MT@if@true
857 \MT@with@babel@and@T{mexican} \MT@if@true
858 \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.)

```
859 \MT@with@package@T{csquotes}{%
860 \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`.

```
861 \MT@if@false
862 \MT@with@package@T{hyperref} \MT@if@true
863 \MT@with@package@T{tex4ht} \MT@if@true
864 \MT@with@package@T{mathastext} \MT@if@true
865 \ifMT@if@MT@restore@p@h\fi
866 \MT@with@package@T{tikz} \MT@tikz@setup
867 }
```

Check again at the end of the preamble.

```
868 </package>
869 \MT@addto@setup{%
870 <*package>
```

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

```
871 \MT@with@package@T{pdfcpot}{%
872 \MT@error{Detected the `pdfcpot' package!\MessageBreak
873 \MT@MT' and `pdfcpot' may not be used together}{%
874 The `pdfcpot' package provides an interface to character protrusion.\MessageBreak
875 So does the `MT@MT' package. Using both packages at the same\MessageBreak
876 time will almost certainly lead to undesired results. Have your choice!}%
877 }%
878 \MT@with@package@T {ledmac} \MT@ledmac@setup
879 \MT@with@package@T {eledmac} \MT@ledmac@setup
880 \MT@with@package@T{reledmac} \MT@ledmac@setup
881 \MT@with@package@T{xunicode} \MT@xunicodetrue
882 </package>
883 <plain> \MT@requires@latex2{
884 \MT@with@package@T{fontspec}{\MT@fontspectrue\MT@fontspec@setup}%
885 <plain> } \relax
886 <*package>
```

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

```
887 \MT@gl@et\MT@setupfont@hook\@empty
888 \ifMT@fontspec
889 \g@addto@macro\MT@setupfont@hook{\MT@font}%
890 \fi
891 \MT@if@false
892 \MT@with@babel@and@T{spanish} \MT@if@true
893 \MT@with@babel@and@T{galician} \MT@if@true
```

```

894 \MT@with@babel@and@T{mexican} \MT@if@true
895 \ifMT@if@
896 \g@addto@macro\MT@setupfont@hook{%
897 \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
898 \fi
899 \MT@with@package@T{csquotes}{%
900 \ifpackage@later{csquotes}{2005/05/11}{%
901 \g@addto@macro\MT@setupfont@hook\@disablequotes
902 }{%
903 \MT@warning@n1{%
904 Should you receive warnings about unknown slot\MessageBreak
905 numbers, try upgrading the `csquotes' package}%
906 }%
907 }%

```

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

```

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

```

The `listings` package makes numbers and letters active,

```

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

```

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

```

934 \g@addto@macro\MT@setupfont@hook{%
935 \catcode`\z@

```

Inside a listing, `\space` is redefined.

```

936 \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.

```

937 \let\lst@ProcessLetter\@empty
938 }%
939 }%

```

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.

```
940 </package>
941 <plain> \MT@requires@latex2{
942   \MT@with@package@T{soul}{%
943     \soulregister\lsstyle 0%
944     \soulregister\textls 1%
945   }%
```

Under plain T<sub>E</sub>X, `soul` doesn't register itself the L<sup>A</sup>T<sub>E</sub>X way, hence we have to use a different test in this case.

```
946 <*plain>
947   {\ifx\SOU@L@\undefined\else
948     \soulregister\lsstyle 0%
949     \soulregister\textls 1%
950   \fi}%
951 </plain>
952 <*package>
953   \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.

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

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

```
971 <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`). But first, we might have to disable stuff when used together with adventurous packages.

```
972 <*pdfTeX-def|xetex-def|luatex-def>
973 \def\MT@setupfont{\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).

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

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

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

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

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

```
982 % \ifx\cf@encoding\cf@encoding\else\@@enc@update\fi
983 }
```

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

```
984 <pdfTeX-def>\MT@requires@pdfTeX6
985 <luaTeX-def>\MT@requires@luaTeX3
986 <pdfTeX-def>|<luaTeX-def> \g@addto@macro\MT@setupfont\MT@tracking}\relax
987 \g@addto@macro\MT@setupfont{%
988 \MT@check@font
989 \ifMT@inlist@
990 <debug>\MT@show@pdfannot2%
991 \else
992 \MT@info{Setting up font `'\MT@font'\on@line}%
993 \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.

```
994 \MT@protrusion
995 <pdfTeX-def>|<luaTeX-def> \MT@expansion
996 }
```

Interword spacing and kerning (pdfTeX 1.40).

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

Disable ligatures (pdfTeX 1.30).

```
1002 <pdfTeX-def>\MT@requires@pdfTeX5{
1003 <pdfTeX-def>|<luaTeX-def>\g@addto@macro\MT@setupfont\MT@noLigatures
1004 <pdfTeX-def>}\relax
1005 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
1006 <debug>\MT@show@pdfannot1%
```

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

```
1007 \MT@register@font
1008 \fi
1009 }
1010 </pdfTeX-def>|<xetex-def>|<luaTeX-def>
```

`\MT@copy@font`  
`\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.

```

1011 <pdfTeX-def|luatex-def>
1012 \let\MT@font@copy@font\relax
1013 <luatex-def>\MT@requires@luatex4{\let\pdfcopyfont\copyfont}\relax
1014 <pdfTeX-def>\MT@requires@pdfTeX7{
1015 \def\MT@font@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@gl@et\MT@font@orig\font@name
1021 \else
1022 \MT@exp@two@c\let\font@name\MT@font@orig
1023 \fi
1024 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
1025 <debug>\MT@dinfo{creating new copy: \MT@font@copy}%

```

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

```

1026 \MT@map@clist@c\MT@active@features{%
1027 \MT@exp@cs\ifx\MT@\@nameuse\MT@abbr@##1}\relax\else
1028 \def\@tempa{##1}%
1029 \MT@exp@cs\MT@map@tlist@c\MT@##1@doc@contexts}\MT@rem@from@list
1030 \fi
1031 }%
1032 \fi
1033 \MT@exp@two@c\let\MT@font\MT@font@copy

```

We only need the font identifier for letterspacing.

```

1034 \let\font@name\MT@font@copy

```

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

```

1035 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
1036 }

```

\MT@rem@from@list

```

1037 \def\MT@rem@from@list#1{%
1038 \MT@exp@cs\ifx\MT@\@tempa @#1font@list}\relax\else
1039 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
1040 \MT@font \csname \MT@\@tempa @#1font@list\endcsname
1041 \fi
1042 }
1043 <pdfTeX-def>\relax
1044 </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@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@series 1045 <package>
\MT@shape 1046 \def\MT@split@name#1/#2/#3/#4/#5/#6\nil{%
1047 \def\MT@encoding{#1}%
\MT@size 1048 \ifMT@fontspec
1049 \edef\MT@family{\MT@scrubfeature#2()\relax}%
1050 \else
1051 \def\MT@family{#2}%
1052 \fi
1053 \def\MT@series {#3}%
1054 \def\MT@shape {#4}%
1055 \def\MT@size {#5}%

```

`\MT@familyalias` Alias family?

```

1056 \MT@ifdefined@n@TF{MT@\MT@family @alias}%
1057 {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
1058 {\let\MT@familyalias\@empty}%
1059 }

```

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

```

\MT@scrubfeatures 1060 \def\MT@scrubfeature#1(#2)#3\relax{#1}
1061 \def\MT@scrubfeatures#1(#2)#3\relax{%
1062 #1%
1063 \ifx\relax#3\relax\else
1064 \MT@scrubfeatures#3\relax
1065 \fi
1066 }

```

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

```

\MT@maybe@do 1067 \newif\ifMT@do
1068 \def\MT@maybe@do#1{%

```

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

```

1069 \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).

```

1070 \MT@dotrue
1071 \edef\@tempa{\csname MT@#1@setname\endcsname}%
1072 \MT@map@clist@n{font,encoding,family,series,shape,size}%
1073 \MT@ifdefined@n@TF{MT@checklist@##1}%
1074 {\csname MT@checklist@##1\endcsname}%
1075 {\MT@checklist@{##1}}%
1076 {#1}%
1077 }%
1078 \else
1079 \MT@dofalse
1080 \fi
1081 \ifMT@do

```

\MT@feat stores the current feature.

```

1082 \def\MT@feat{#1}%
1083 \csname MT@set@#1@codes\endcsname
1084 \else
1085 \MT@ifstreq{#1}{tr}%
1086 {\let\MT@info@nottracking\MT@info@nottracking@}%
1087 {\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}%
1088 \fi
1089 }

```

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

```

\MT@info@nottracking@ 1090 \let\MT@info@nottracking\relax
1091 \def\MT@info@nottracking@{\MT@vinfo{... No tracking}}

```

\MT@dinfo@list

```

1092 <debug>\def\MT@dinfo@list#1#2#3{\MT@dinfo@n1}{\@nameuse{MT@abbr@#1}: #2
1093 <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).

```

1094 \def\MT@checklist@#1#2{%
1095 <!debug> \MT@ifdefined@n@T
1096 <debug> \MT@ifdefined@n@TF
1097 {MT@#2list@#1@\@tempa}{%

```

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

```

1098 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
1099 \csname MT@#1\expandafter\endcsname
1100 \csname MT@#2list@#1@\@tempa\endcsname
1101 \ifMT@inlist@
1102 <debug>\MT@dinfo@list{#2}{#1}{in}%
1103 \MT@dotrue
1104 \else
1105 <debug>\MT@dinfo@list{#2}{#1}{not in}%
1106 \MT@dofalse
1107 \expandafter\MT@clist@break
1108 \fi
1109 }%

```

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.

```

1110 <debug> {\MT@dinfo@list{#2}{#1}{}}%
1111 }

```

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

```

1112 \def\MT@checklist@family#1{%
1113 <!debug> \MT@ifdefined@n@T
1114 <debug> \MT@ifdefined@n@TF
1115 {MT@#1list@family@\@tempa}{%
1116 \MT@exp@two@n\MT@in@clist
1117 \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
1118 \ifMT@inlist@
1119 <debug>\MT@dinfo@list{#1}{family}{in}%
1120 \MT@dotrue
1121 \else
1122 <debug>\MT@dinfo@list{#1}{family}{not in}%
1123 \MT@dofalse
1124 \ifx\MT@familyalias\empty \else
1125 \MT@exp@two@n\MT@in@clist
1126 \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
1127 \ifMT@inlist@
1128 <debug> \MT@dinfo@list{#1}{family alias}{in}%
1129 \MT@dotrue
1130 <debug>\else\MT@dinfo@list{#1}{family alias}{not in}%

```

```

1131     \fi
1132     \fi
1133     \fi
1134     \ifMT@do \else
1135         \expandafter\MT@clist@break
1136     \fi
1137 }%
1138 <debug> {\MT@info@list{#1}{family}{}}%
1139 }

```

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

```

1140 \def\MT@checklist@size#1{%
1141 <!debug> \MT@ifdefined@n@T
1142 <debug> \MT@ifdefined@n@TF
1143     {MT#1list@size@tempa}%
1144     \MT@exp@cs\MT@in@rlist{MT#1list@size@tempa}%
1145     \ifMT@inlist@
1146 <debug> \MT@info@list{#1}{size}{in}%
1147     \MT@dotrue
1148     \else
1149 <debug> \MT@info@list{#1}{size}{not in}%
1150     \MT@dofalse
1151     \expandafter\MT@clist@break
1152     \fi
1153 }%
1154 <debug> {\MT@info@list{#1}{size}{}}%
1155 }

```

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

```

1156 \def\MT@checklist@font#1{%
1157 <!debug> \MT@ifdefined@n@T
1158 <debug> \MT@ifdefined@n@TF
1159     {MT#1list@font@tempa}%

```

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

```

1160     \edef@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1161     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1162     \@tempb \csname MT#1list@font@tempa\endcsname
1163     \ifMT@inlist@
1164 <debug> \MT@info@list{#1}{font}{in}%
1165     \expandafter\MT@clist@break
1166     \else
1167 <debug> \MT@info@list{#1}{font}{not in}%
1168     \MT@dofalse
1169     \fi
1170 }%
1171 <debug> {\MT@info@list{#1}{font}{}}%
1172 }

```

### 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`.

```

1173 \newif\ifMT@nofamily
1174 </package>

```

`\MT@protrusion` Set up for protrusion?

```

1175 <*pdfTeX-def|xetex-def|luatex-def>
1176 \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.

```
1177 \def\MT@set@pr@codes{%
1178   \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).

```
1179   \MT@if@list@exists{%
1180     \ifMT@nofamily
1181       \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
1182         \MT@info@n@l{Loading generic protrusion settings for font family\MessageBreak
1183           ~\MT@family' (encoding: \MT@encoding).\MessageBreak
1184           For optimal results, create family-specific settings.\MessageBreak
1185           See the microtype manual for details}%
1186         \MT@gl@et@nc{\MT@encoding-\MT@family-settings}\@empty
1187       }%
1188     \fi
1189   \MT@get@font@dimen@six{%
1190     \MT@get@opt
1191     \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

```
1192   \MT@get@inh@list
```

Set an input encoding?

```
1193   \MT@set@inputenc{c}%
```

Load additional lists?

```
1194   \MT@load@list\MT@pr@c@name
1195   \MT@set@listname
```

Load the main list.

```
1196   \MT@let@cn@tempc{\MT@pr@c@\MT@pr@c@name}%
1197   \expandafter\MT@set@codes\@tempc,\relax,%
1198 } \MT@reset@pr@codes
1199 }
```

`\MT@get@font@dimen@six` If `\fontdimen 6` is zero, character protrusion, spacing, kerning and tracking won't work, and we can skip the settings (for example, the `dsfont` and `fourier` fonts don't specify this dimension; this is probably a bug in the fonts).

`\MT@dimen@six`

```
1200 \def\MT@get@font@dimen@six{%
1201   \ifnum\fontdimen6\MT@font=\z@
1202     \MT@warning@n@l{%
1203       Font ~\MT@font' does not specify its\MessageBreak
1204       \@backslashchar fontdimen 6 (width of an 'em')! Therefore,\MessageBreak
1205       \@nameuse{\MT@abbr@\MT@feat} will not work with this font}%
1206     \expandafter\@gobble
1207   \else
1208     \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1209     \expandafter\@firstofone
1210   \fi
1211 }
```

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

```
1212 \def\MT@set@all@pr#1#2{%
1213   (debug)\MT@info@n@l{3}{-- lp/rp: setting all to #1/#2}%
1214   \let\MT@temp\@empty
1215   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1}}%
1216   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2}}%
1217   \MT@do@font\MT@temp
1218 }
```

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

`\MT@reset@pr@codes`

\microtypecontext if necessary.

```
1219 \def\MT@reset@pr@codes@{\MT@set@all@pr@z@z@}
1220 \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.

```
1221 \def\MT@the@pr@code{\@tempcntb}
1222 <pdf-tex-def|luatex-def>
1223 <pdf-tex-def>\MT@requires@pdf-tex6
1224 <luatex-def>\MT@requires@luatex3
1225 {\def\MT@the@pr@code@tr{%
1226   \numexpr\@tempcntb+\MT@letterspace@/2\relax
1227 }
1228 }\relax
1229 </pdf-tex-def|luatex-def>
```

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

```
1230 \def\MT@set@codes#1,{%
1231   \ifx\relax#1\@empty\else
1232     \MT@split@codes #1==\relax
1233     \expandafter\MT@set@codes
1234   \fi
1235 }
```

\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.

```
1236 \def\MT@split@codes#1=#2=#3\relax{%
1237   \def\@tempa{#1}%
1238   \ifx\@tempa\@empty \else
1239     \MT@get@slot
1240     <pdf-tex-def|luatex-def> \ifnum\MT@char > \m@ne
1241     <xetex-def> \ifx\MT@char\@empty \else
1242       \MT@get@char@unit
1243       \csname MT@\MT@feat @split@val\endcsname#2\relax
1244     \fi
1245   \fi
1246 }
```

\MT@pr@split@val

```
1247 \def\MT@pr@split@val#1,#2\relax{%
1248   \def\@tempb{#1}%
1249   \MT@ifempty\@tempb\relax{%
1250     \MT@scale@to@em
1251     \lcode\MT@font\MT@char=\MT@the@pr@code
1252     <debug>\MT@info@n1{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char\space: [#1]}%
1253   }%
1254   \def\@tempb{#2}%
1255   \MT@ifempty\@tempb\relax{%
1256     \MT@scale@to@em
1257     \rcode\MT@font\MT@char=\MT@the@pr@code
1258     <debug>\MT@info@n1{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char\space: [#2]}%
1259   }%
```

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

```
1260 \MT@ifdefined@c@T\MT@pr@inh@name{%
1261   \MT@ifdefined@nT\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1262     \MT@exp@cs\MT@map@tlist@c
1263     {\MT@inh@\MT@pr@inh@name @\MT@char @}%
1264     \MT@set@pr@heirs
1265   }%
1266 }
```

1267 }

`\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 `\lcode` resp. `\rcode`, since this would disallow protrusion factors larger than the character width (since `\[1r]pcode`’s limit is 1000). Now, the maximum protrusion is 1 em of the font.

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

1268 `<pdfTeX-def>\MT@requires@pdfTeX3{`1269 `\def\MT@scale@to@em{%`1270 `\@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`.

1271 `\MT@scale\@tempcntb \@tempb \MT@dimen@six`1272 `\ifnum\@tempcntb=\z@ \else`1273 `\MT@scale@factor`1274 `\fi`

1275 }

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

1276 `\def\MT@get@charwd{%`1277 `<pdfTeX-def>`1278 `^^X \MT@count=\fontcharwd\MT@font\MT@char\relax`1279 `^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%`1280 `^^Q \MT@count=\wd\z@`1281 `</pdfTeX-def>`1282 `<LaTeX-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).

1283 `<xetex-def>`1284 `\ifnum\MT@char@<\z@`1285 `\setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}%`1286 `\MT@count=\wd\z@`1287 `\else`1288 `\MT@count=\fontcharwd\MT@font\MT@char@\relax`1289 `\fi`1290 `</xetex-def>`1291 `\ifnum\MT@count=\z@ \MT@info@missing@char \fi`

1292 }

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

1293 `<pdfTeX-def>`1294 `\MT@requires@pdfTeX6{`1295 `\g@addto@macro\MT@get@charwd{%`1296 `\MT@ifdefined@cT\MT@letterspace@`1297 `{\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%`1298 `}`1299 `}\relax`1300 `}}`

No adjustment with versions 0.14f and 0.14g.

```
1301 \def\MT@scale@to@em{%
1302   \MT@count=\@tempb\relax
1303   \ifnum\MT@count=\z@ \else
1304     \MT@scale@factor
1305   \fi
1306 }
```

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

```
1307 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1308 }
1309 </pdfTeX-def>
1310 </pdfTeX-def|xetex-def|luatex-def>
```

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

```
1311 <{*package}>
1312 \def\MT@get@font@dimen#1{%
1313   \ifnum\fontdimen#1\MT@font=\z@
1314     \MT@warning@n1{Font `'\MT@@font' does not specify its\MessageBreak
1315       \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1316       You should use a different `unit' for \MT@curr@list@name}%
1317   \else
1318     \MT@count=\fontdimen#1\MT@font
1319   \fi
1320 }
```

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

```
1321 \def\MT@info@missing@char{%
1322   \MT@info@n1{Character `'\the\MT@toks'
1323   ^^X   \iffontchar\MT@font\MT@char@
1324     has a width of 0pt
1325   ^^X   \else is missing\fi
1326   ^^Q   \MessageBreak (it's probably missing)
1327   \MessageBreak in font `'\MT@@font'. \MessageBreak
1328   Ignoring protrusion settings for this character}%
1329 }
```

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

```
1330 \def\MT@scale@factor{%
1331   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1332     \expandafter\MT@scale\expandafter \@tempcntb
1333     \csname MT@\MT@feat @factor@\endcsname \@m
1334   \fi
1335   \ifnum\@tempcntb>\csname MT@\MT@feat @max@\endcsname\relax
1336     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1337   \else
1338     \ifnum\@tempcntb<\csname MT@\MT@feat @min@\endcsname\relax
1339       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1340     \fi
1341   \fi
1342 }
```

`\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.

```
1343 \def\MT@warn@code@too@large#1{%
1344   \@tempcnta=#1\relax
1345   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1346     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1347     \@m \csname MT@\MT@feat @factor@\endcsname
1348   \fi
1349   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1350   \MT@warning@n1{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1351     is too large for character \MessageBreak
```

```

1352   ~\the\MT@toks' in \MT@curr@list@name.\MessageBreak
1353   Setting it to the maximum of \number\@tempcnta}%
1354   \@tempcntb=#1\relax
1355 }

\MT@get@opt    The optional argument to the configuration commands (except for \SetExpansion,
               which is being dealt with in \MT@get@ex@opt).

1356 \def\MT@get@opt{%
1357   \MT@set@listname

\MT@pr@factor@    Apply a factor?

\MT@sp@factor@ 1358 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1359 \MT@let@nn{MT@\MT@feat @factor@}
1360   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1361   \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1362   \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1363   }%
1364   \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1365   }%

\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@ 1366 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1367   \MT@let@nn{MT@\MT@feat @unit@}%
1368   {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1369   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1370   \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1371   relative to character widths}%
1372   \else
1373   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1374   \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1375   relative to width of space}%
1376   \fi
1377   \fi
1378   }%
1379   \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1380   }%

\MT@get@space@unit  The codes are either relative to character widths, or to a fixed width. For spacing
\MT@get@char@unit  and kerning lists, they may also be relative to the width of the interword glue. Only
                   the setting from the top list will be taken into account.

1381   \let\MT@get@char@unit\relax
1382   \let\MT@get@space@unit\@gobble
1383   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\@empty
1384   \let\MT@get@char@unit\MT@get@charwd
1385   \else
1386   \MT@exp@cs@ifx{MT@\MT@feat @unit@}\m@ne
1387   \let\MT@get@space@unit\MT@get@font@dimen
1388   \else
1389   \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1390   \fi
1391   \fi

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

1392   \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
1393   \csname MT@preset@\MT@feat\endcsname
1394   \MT@let@nc{MT@reset@\MT@feat @codes}\relax
1395   }%
1396 }

\MT@get@unit    If unit contains an em or ex, we use the corresponding \fontdimen to obtain the
\MT@get@unit@    real size. Simply converting the em into points might give a wrong result, since
                   the font probably isn't set up yet, so that these dimensions haven't been updated,

```

either.

```

1397 \def\MT@get@unit#1{%
1398   \expandafter\MT@get@unit@#1 e!\@nil
1399   \ifx\x\@empty\else\let#1\x\fi
1400   \@defaultunits\@tempdima#1 pt\relax\@nnil
1401   \ifdim\@tempdima=\z@
1402     \MT@warning@nl{%
1403       Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
1404       width. Setting factors of list \@nameuse{MT@\MT@feat @c@name}'\MessageBreak
1405       relative to character widths instead}%
1406     \let#1\@empty
1407     \let\MT@get@char@unit\MT@get@charwd
1408   \else
1409     \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
1410       to \the\@tempdima}%
1411     \MT@count=\@tempdima\relax
1412   \fi
1413 }
1414 \def\MT@get@unit@#1e#2#3\@nil{%
1415   \ifx\#3\@empty\else
1416     \if m#2%
1417       \edef\x{#1\fontdimen6\MT@font}%
1418     \else
1419       \if x#2%
1420         \edef\x{#1\fontdimen5\MT@font}%
1421       \fi
1422     \fi
1423   \fi
1424 }

```

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

```
1425 \def\MT@set@inputenc#1{%
```

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

```

1426   \def\MT@cat{#1}%
1427   \edef\@tempa{MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}%
1428   \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1429 }

```

\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.

```

1430 \MT@addto@setup{%
1431   \ifpackageloaded{inputenc}{%
1432     \ifpackageafter{inputenc}{2006/02/22}{%
1433       \def\MT@set@inputenc@{%
1434         \MT@ifstreqlinputencodingname{\csname\@tempa\endcsname}\relax
1435         \MT@load@inputenc
1436       }%
1437     }%
1438     \let\MT@set@inputenc@\MT@load@inputenc
1439   }%
1440 }%
1441 \def\MT@set@inputenc@{%
1442   \MT@warning@nl{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1443     \MessageBreak package isn't loaded. Ignoring input encoding}%
1444 }%
1445 }%
1446 }

```

\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.

```

1447 \def\MT@load@inputenc{%
1448   \MT@cfg@catcodes
1449   (debug)\MT@dinfo@nl{1}{loading input encoding: \@nameuse{\@tempa}}%

```

```
1450 \inputencoding{\@nameuse{\@tempa}}%
1451 }
1452 /package
```

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

```
1453 (*pdfTeX-def|xetex-def|luatex-def)
1454 \def\MT@set@pr@heirs#1{%
1455   \lcode\MT@font #1 =\lcode\MT@font\MT@char\relax
1456   \rcode\MT@font #1 =\rcode\MT@font\MT@char\relax
1457   <debug>\MT@dinfoln{2}{-- heir of \MT@char: #1}%
1458   <debug>\MT@dinfoln{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char\space/%
1459   <debug>                                     \number\rcode\MT@font\MT@char\space}%
1460 }
```

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

```
\MT@preset@pr@ 1461 \def\MT@preset@pr{%
1462   \expandafter\expandafter\expandafter\MT@preset@pr@
1463   \cscname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1464 }
1465 \def\MT@preset@pr@#1,#2\@nil{%
1466   \ifx\MT@pr@unit@\@empty
1467     \MT@warn@preset@tewidth{pr}%
1468     \let\MT@preset@aux\MT@preset@aux@factor
1469   \else
1470     \def\MT@preset@aux{\MT@preset@aux@space2}%
1471   \fi
1472   \MT@i fempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1473   \MT@i fempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1474   \MT@set@all@pr\@tempa\@tempb
1475 }
```

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

```
\MT@preset@aux@factor 1476 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1477   \@tempcntb=#1\relax
1478   \MT@scale@factor
1479   \edef#2{\number\@tempcntb}%
1480 }
1481 \def\MT@preset@aux@space#1#2#3{%
1482   \def\@tempb{#2}%
1483   \MT@get@space@unit#1%
1484   \MT@scale@to@em
1485   \edef#3{\number\@tempcntb}%
1486 }
```

`\MT@warn@preset@tewidth`

```
1487 \def\MT@warn@preset@tewidth#1{%
1488   \MT@warning@nl{%
1489     Cannot preset characters relative to their widths\MessageBreak
1490     for \@nameuse{MT@abbr@#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1491     \MessageBreak relative to lem instead}%
1492 }
1493 /pdfTeX-def|xetex-def|luatex-def
```

### 14.2.2 Expansion

`\MT@expansion` Set up for expansion?

```
1494 <pdfTeX-def|luatex-def>
1495 \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).

```

1496 \def\MT@set@ex@codes@{%
1497   \MT@if@list@exists{%
1498     \MT@get@ex@opt
1499     \let\MT@get@char@unit\relax
1500     \MT@reset@ef@codes
1501     \MT@get@inh@list
1502     \MT@set@inputenc{c}%
1503     \MT@load@list\MT@ex@cc@name
1504     \MT@set@listname
1505     \MT@let@cn\@tempc{MT@ex@cc@MT@ex@cc@name}%
1506     \expandafter\MT@set@codes\@tempc,\relax,%
1507     \MT@expandfont
1508   }\relax
1509 }
1510 (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.

```

1511 (package)\newif\ifMT@nonselected
1512 (*pdftex-def|luatex-def)
1513 \def\MT@set@ex@codes@n{%
1514   \MT@nonselectedtrue
1515   \MT@if@list@exists
1516   \MT@get@ex@opt
1517   {%
1518     \let\MT@stretch@ \MT@stretch
1519     \let\MT@shrink@ \MT@shrink
1520     \let\MT@step@ \MT@step
1521 (pdftex-def) \let\MT@auto@ \MT@auto
1522     \let\MT@ex@factor@\MT@ex@factor
1523   }%
1524   \MT@reset@ef@codes
1525   \MT@expandfont
1526   \MT@nonselectedfalse
1527 }

```

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

```
1528 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

`\MT@expandfont` Expand the font.

```

1529 (luatex-def)\MT@requires@luatex4{\let\pdfontexpand\expandglyphsinfont}\relax
1530 \def\MT@expandfont{%
1531   \pdfontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1532 }

```

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

`\MT@reset@ef@codes@`

```

1533 \def\MT@set@all@ex#1{%
1534 (debug)\MT@dinfo@n{3}{-- ex: setting all to \number#1}%
1535   \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1536 }
1537 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}

```

`\MT@reset@ef@codes` However, this is only necessary for pdf<sub>T</sub>E<sub>X</sub> versions prior to 1.20, or Lua<sub>T</sub>E<sub>X</sub> < 0.90 (actually, I think, 0.87).

```

1538 (pdftex-def)\MT@requires@pdftex4
1539 (luatex-def)\MT@requires@luatex5
1540 {
1541   \def\MT@reset@ef@codes{%

```

```

1542 \ifnum\MT@ex@factor@=\@m \else
1543 \MT@reset@ef@codes@
1544 \fi
1545 }
1546 }{
1547 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1548 }

```

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

```

1549 \def\MT@ex@split@val#1\relax{%
1550 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1551 \ifnum\MT@ex@factor@=\@m \else
1552 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1553 \fi
1554 \ifnum\@tempcntb > \MT@ex@max
1555 \MT@warn@ex@too@large\MT@ex@max
1556 \else
1557 \ifnum\@tempcntb < \MT@ex@min
1558 \MT@warn@ex@too@large\MT@ex@min
1559 \fi
1560 \fi
1561 \efcode\MT@font\MT@char=\@tempcntb
1562 (debug)\MT@dinfol{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1563 \MT@ifdefined@c@T\MT@ex@inh@name{%
1564 \MT@ifdefined@n@T\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1565 \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1566 }%
1567 }%
1568 }

```

\MT@warn@ex@too@large

```

1569 \def\MT@warn@ex@too@large#1{%
1570 \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1571 character\MessageBreak `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1572 Setting it to the maximum of \number#1}%
1573 \@tempcntb=#1\relax
1574 }

```

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

```

\MT@ex@factor@ 1575 \def\MT@get@ex@opt{%
\MT@stretch@ 1576 \MT@set@listname
1577 \MT@ifdefined@n@TF\MT@ex@c@\MT@ex@c@name @factor}{%
\MT@shrink@ 1578 \MT@let@cn\MT@ex@factor@\MT@ex@c@\MT@ex@c@name @factor}%
\MT@step@ 1579 \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1580 }{%
1581 \let\MT@ex@factor@\MT@ex@factor
1582 }%
1583 \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1584 \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1585 \MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}%
1586 (pdfTeX-def) \def\@tempa{autoexpand}%
1587 (pdfTeX-def) \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ E\nelse Dis\fi abling automatic expansion}%
1588 \MT@ifdefined@n@T\MT@ex@c@\MT@ex@c@name @preset}{%
1589 \MT@preset@ex
1590 \let\MT@reset@ef@codes\relax
1591 }%
1592 }

```

\MT@get@ex@opt@

```

1593 \def\MT@get@ex@opt@#1#2{%
1594 \MT@ifdefined@n@TF\MT@ex@c@\MT@ex@c@name @#1}{%

```

```

1595 \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1596 \MT@vinfo{... : #2}%
1597 }{%
1598 \MT@let@nn{MT@#1@}{MT@#1}%
1599 }%
1600 }

```

`\MT@set@ex@heirs`

```

1601 \def\MT@set@ex@heirs#1{%
1602 \efcode\MT@font#1=\efcode\MT@font\MT@char
1603 <debug>\MT@dinfo@n1{2}{-- heir of \MT@char: #1}%
1604 <debug>\MT@dinfo@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1605 }

```

`\MT@preset@ex`

```

1606 \def\MT@preset@ex{%
1607 \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1608 \MT@scale@factor
1609 \MT@set@all@ex\@tempcntb
1610 }
1611 </pdfTeX-def|luatex-def>

```

### 14.2.3 Interword spacing (glue)

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

```

1612 <*pdfTeX-def>
1613 \MT@requires@pdfTeX6{
1614 \def\MT@spacing{\MT@maybe@do{sp}}

```

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

```

1615 \def\MT@set@sp@codes{%
1616 \MT@if@list@exists{%
1617 \MT@get@font@dimen@six{%
1618 \MT@get@opt
1619 \MT@reset@sp@codes
1620 \MT@get@inh@list
1621 \MT@set@inputenc{c}%
1622 \MT@load@list\MT@sp@c@name
1623 \MT@set@listname
1624 \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
1625 \expandafter\MT@set@codes\@tempc,\relax,}%
1626 }\MT@reset@sp@codes
1627 }

```

`\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).

```

1628 \def\MT@sp@split@val#1,#2,#3\relax{%
1629 \def\@tempb{#1}%
1630 \MT@ifempty\@tempb\relax{%
1631 \MT@get@space@unit2%
1632 \MT@scale@to@em
1633 \knbscode\MT@font\MT@char=\@tempcntb
1634 <debug>\MT@dinfo@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1635 }%
1636 \def\@tempb{#2}%
1637 \MT@ifempty\@tempb\relax{%
1638 \MT@get@space@unit3%
1639 \MT@scale@to@em
1640 \stbscode\MT@font\MT@char=\@tempcntb
1641 <debug>\MT@dinfo@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1642 }%
1643 \def\@tempb{#3}%
1644 \MT@ifempty\@tempb\relax{%

```

```

1645 \MT@get@space@unit4%
1646 \MT@scale@to@em
1647 \shbscode\MT@font\MT@char=\@tempcntb
1648 (debug)\MT@edinfo@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1649 }%
1650 \MT@ifdefined@c@T\MT@sp@inh@name{%
1651 \MT@ifdefined@nT\MT@inh@\MT@sp@inh@name @\MT@char @}{%
1652 \MT@exp@cs\MT@map@tlist@c\MT@inh@\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
1653 }%
1654 }%
1655 }

\MT@set@sp@heirs

1656 \def\MT@set@sp@heirs#1{%
1657 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1658 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1659 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1660 (debug)\MT@edinfo@n1{2}{-- heir of \MT@char: #1}%
1661 (debug)\MT@edinfo@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1662 (debug) \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1663 }

\MT@set@all@sp

\MT@reset@sp@codes 1664 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1665 (debug)\MT@edinfo@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1666 \let\MT@temp@empty
1667 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1668 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1669 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1670 \MT@do@font\MT@temp
1671 }
1672 \def\MT@reset@sp@codes@\MT@set@all@sp\z@\z@\z@
1673 \let\MT@reset@sp@codes\relax

\MT@preset@sp

\MT@preset@sp@ 1674 \def\MT@preset@sp{%
1675 \expandafter\expandafter\expandafter\MT@preset@sp@
1676 \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1677 }
1678 \def\MT@preset@sp@#1.#2.#3\@nil{%
1679 \ifx\MT@sp@unit@\@empty
1680 \MT@warn@preset@twidth{sp}%
1681 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1682 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1683 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1684 \else
1685 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1686 \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1687 \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1688 \fi
1689 \MT@set@all@sp@\@tempa\@tempc\@tempb
1690 }
1691 }\relax

```

#### 14.2.4 Additional kerning

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

```

1692 \MT@requires@pdftex6{
1693 \def\MT@kerning{\MT@maybe@do{kn}}

```

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

```

1694 \def\MT@set@kn@codes{%
1695 \MT@if@list@exists{%

```

```

1696 \MT@get@font@dimen@six{%
1697 \MT@get@opt
1698 \MT@reset@kn@codes
1699 \MT@get@inh@list
1700 \MT@set@inputenc{c}%
1701 \MT@load@list\MT@kn@c@name
1702 \MT@set@listname
1703 \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1704 \expandafter\MT@set@codes\@tempc,\relax,%
1705 }\MT@reset@kn@codes
1706 }

```

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

```

1707 \def\MT@kn@split@val#1,#2\relax{%
1708 \def\@tempb{#1}%
1709 \MT@ifempty\@tempb\relax{%
1710 \MT@get@space@unit2%
1711 \MT@scale@to@em
1712 \knbccode\MT@font\MT@char=\@tempcntb
1713 debug\MT@dinfoln{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
1714 }%
1715 \def\@tempb{#2}%
1716 \MT@ifempty\@tempb\relax{%
1717 \MT@get@space@unit2%
1718 \MT@scale@to@em
1719 \knaccode\MT@font\MT@char=\@tempcntb
1720 debug\MT@dinfoln{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1721 }%
1722 \MT@ifdefined@c@T\MT@kn@inh@name{%
1723 \MT@ifdefined@nT{MT@inh@\MT@kn@inh@name @\MT@char @}{%
1724 \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1725 }%
1726 }%
1727 }

```

\MT@set@kn@heirs

```

1728 \def\MT@set@kn@heirs#1{%
1729 \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1730 \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1731 debug\MT@dinfoln{2}{-- heir of \MT@char: #1}%
1732 debug\MT@dinfoln{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
1733 debug \number\knaccode\MT@font\MT@char}%
1734 }

```

\MT@set@all@kn

```

\MT@reset@kn@codes 1735 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1736 debug\MT@dinfoln{3}{-- knac/knbc: setting all to #1/#2}%
1737 \let\MT@temp\@empty
1738 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbccode\MT@font\@tempcnta=#1\relax}}%
1739 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
1740 \MT@do@font\MT@temp
1741 }
1742 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1743 \let\MT@reset@kn@codes\relax

```

\MT@preset@kn

```

\MT@preset@kn@ 1744 \def\MT@preset@kn{%
1745 \expandafter\expandafter\expandafter\MT@preset@kn@
1746 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1747 }
1748 \def\MT@preset@kn@#1,#2\@nil{%
1749 \ifx\MT@kn@unit@\@empty
1750 \MT@warn@preset@twidth{kn}%
1751 \let\MT@preset@aux\MT@preset@aux@factor
1752 \else

```

```

1753 \def\MT@preset@aux{\MT@preset@aux@space2}%
1754 \fi
1755 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1756 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1757 \MT@set@all@kn\@tempa\@tempb
1758 }
1759 }\relax
1760 </pdfTeX-def>

```

### 14.2.5 Tracking

This only works with pdfTeX 1.40 or LuaTeX 0.62.

```

1761 <*pdfTeX-def|luatex-def>
1762 <pdfTeX-def>\MT@requires@pdfTeX6
1763 <luatex-def>\MT@requires@luatex3
1764 {

```

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

```

\MT@tracking@
\MT@tracking@
\MT@tr@font@list 1765 \let\MT@tr@font@list\@empty
1766 \def\MT@tracking@{%
1767 \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1768 \ifMT@inlist@else
1769 \MT@maybe@do{tr}%
1770 \ifMT@do@else
1771 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1772 \fi
1773 \fi
1774 }
1775 </pdfTeX-def|luatex-def>
1776 <pdfTeX-def|luatex-def>\let\MT@tracking
1777 <pdfTeX-def|luatex-def> \MT@tracking@
1778 <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.

```

1779 <*pdfTeX-def|luatex-def|letterspace>
1780 \def\MT@set@tr@codes{%
1781 <*pdfTeX-def|luatex-def>
1782 \MT@vinfo{Tracking font '\MT@font'\on@line}%
1783 \MT@get@font@dimen@six{%
1784 \MT@if@list@exists
1785 \MT@get@tr@opt
1786 \relax
1787 </pdfTeX-def|luatex-def>
1788 \MT@ifdefined@c@TF\MT@letterspace@relax{\let\MT@letterspace@\MT@letterspace}%
1789 \ifnum\MT@letterspace@=\z@

```

Zero tracking requires special treatment.

```

1790 \MT@set@tr@zero
1791 \else
1792 <pdfTeX-def|luatex-def> \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

```

1793 \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.

```

1794 \xdef\MT@lsfont{\csname\expandafter\string\font@name

```

```

1795             /\number\MT@letterspace@ 1s\endcsname}%
1796     \expandafter\ifx\MT@1sfont\relax
1797 <debug>\MT@info@n1{1}{... new letterspacing instance}%

```

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

```
1798     \MT@get@1s@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.

```

1799 <*luatex-def|letterspace>
1800     \MT@if@fontspec@font{%
1801 <luatex-def&debug>\MT@info@n1{1}{... fontspec font: \MessageBreak
1802 <luatex-def&debug>         \expandafter\fontname\font@name}%
1803         \ifnum\MT@letterspace@<z@\def\MT@minus{-}\else\let\MT@minus\empty\fi
1804         \global\expandafter\font\MT@1sfont=%
1805         \expandafter\MT@exp@two@c\expandafter\MT@1s@fontspec@font
1806         \expandafter\fontname\expandafter\font@name\space \@nil
1807     }{%
1808 </luatex-def|letterspace>
1809 <luatex-def&debug>\MT@info@n1{1}{... legacy font}%
1810     \global\expandafter\letterspacefont\MT@1sfont\font@name\MT@letterspace@
1811 <luatex-def|letterspace>     }%

```

Scale interword spacing (not configurable in letterspace).

```

1812 <*pdfTeX-def|luatex-def>
1813     \MT@ifdefined@c@TF\MT@tr@ispace
1814     {\let\@tempa\MT@tr@ispace}%
1815     {\edef\@tempa{\MT@letterspace@*,,}}%
1816     \MT@ifdefined@c@TF\MT@tr@ospace
1817     {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1818     {\edef\@tempa{\@tempa,,,}}%
1819     \expandafter\MT@tr@set@ospace\@tempa,%
1820 </pdfTeX-def|luatex-def>
1821 <*letterspace>
1822     % spacing = {<letterspace amount>*,,}
1823     \fontdimen2\MT@1sfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1824     * \fontdimen2\MT@1sfont/1000\relax
1825 </letterspace>

```

Adjust outer kerning (microtype only).

```

1826 <*pdfTeX-def|luatex-def>
1827     \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1828     \expandafter\MT@tr@set@okern\@tempa,%

```

Disable ligatures (not configurable in letterspace).

```

1829     \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1830 </pdfTeX-def|luatex-def>
1831 <*letterspace>
1832     % no ligatures = {f}
1833     \tagcode\MT@1sfont`f=\m@ne
1834 </letterspace>

```

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

```

1835 <luatex-def|letterspace>     \MT@if@fontspec@font\relax{%
1836 <debug>\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%
1837     \MT@do@font{\lpcode\MT@1sfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1838     \rccode\MT@1sfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1839     \let\MT@the@pr@code\MT@the@pr@code@tr
1840 <luatex-def|letterspace>     }%
1841     \fi

```

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

```
1842 \aftergroup\MT@set@lsfont
1843 (pdfTeX-def|LaTeX-def) \let\MT@font\MT@lsfont
1844 (LaTeX-def) \MT@if@fontspec@font\MT@font\relax
```

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

```
\MT@curr@ls 1845 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1846 \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.

```
1847 (*pdfTeX-def|LaTeX-def)
1848 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1849 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1850 \MT@tr@outer@l
1851 (/pdfTeX-def|LaTeX-def)
```

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

```
1852 \ifx\MT@ls@adjust\empty
1853 (letterspace) % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1854 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1855 \MT@ls@outer@k
```

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

```
1856 (*pdfTeX-def|LaTeX-def)
1857 \else
1858 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1859 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1860 \ifdim\MT@outer@kern=\z@\else \MT@ls@outer@k \fi
1861 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1862 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1863 (/pdfTeX-def|LaTeX-def)
1864 (*letterspace)
1865 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1866 \MT@afteraftergroup{%
1867 \MT@set@curr@ok
1868 \noexpand\MT@ls@outer@k
1869 }%
1870 (/letterspace)
1871 \fi
1872 (*pdfTeX-def|LaTeX-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).

```
1873 \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.

```
1874 \MT@afteraftergroup{%
1875 \MT@set@curr@os
1876 \MT@set@curr@ok
1877 \noexpand\MT@tr@outer@r
1878 }%
1879 (/pdfTeX-def|LaTeX-def)
1880 \fi
1881 (pdfTeX-def|LaTeX-def) }%
1882 }
```

`\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.)

```

1883 \def\MT@afteraftergroup#1{%
1884 <!letterspace> \MT@maybe@gobble@with@tikz{%
1885   \MT@ifdefined@nTF{MT@aftergroup@number\currentgrouplevel}\relax{%
1886     \MT@exp@cs\xdef{MT@aftergroup@number\currentgrouplevel}%
1887     {\MT@exp@cs\MT@gl@et{MT@aftergroup@number\currentgrouplevel}\noexpand\@undefined#1}%
1888     \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
1889     {MT@aftergroup@number\currentgrouplevel}%
1890   }%
1891 <!letterspace> }%
1892 }
1893 </pdfTeX-def|luatex-def|letterspace>

```

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

`\MT@ls@fontspec@font`

```

1894 <*luatex-def|letterspace>
1895 \def\MT@ls@fontspec@colon#1:#2:#3:#4@nil{\ifx\#3\#1:#2\else#1:#2:#3\fi}
1896 \def\MT@ls@fontspec@font#1 #2@nil{%
1897   "\MT@ls@fontspec@colon#1:::\relax@nil
1898   kernfactor=\MT@minus \ifnum\MT@letterspace@=1000 1\else 0.%
1899     \ifnum\MT@minus\MT@letterspace@<100 0\fi
1900     \ifnum\MT@minus\MT@letterspace@<10 0\fi
1901     \number\MT@minus\MT@letterspace@ \fi;"
1902   \ifx\#2\ at \f@size pt\else#2\fi\relax
1903 }
1904 </luatex-def|letterspace>

```

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

```

1905 <*pdfTeX-def|luatex-def>
1906 \def\MT@get@tr@opt{%
1907   \MT@set@listname
1908   \MT@ifdefined@nT{MT@tr@cc@MT@tr@cc@name}{%
1909     \MT@let@cn\MT@letterspace{MT@tr@cc@MT@tr@cc@name}%

```

`\MT@tr@unit@` Different unit?

```

1910   \MT@ifdefined@nT{MT@tr@cc@MT@tr@cc@name @unit}{%
1911     \MT@let@cn\MT@tr@unit@{MT@tr@cc@MT@tr@cc@name @unit}%
1912     \ifdim\MT@tr@unit@=1em
1913       \let\MT@tr@unit@\@undefined
1914     \else
1915       \MT@let@cn@tempb{MT@tr@cc@MT@tr@cc@name}%
1916       \MT@get@unit\MT@tr@unit@
1917       \let\MT@tr@factor@\@m
1918       \MT@scale@to@em
1919       \edef\MT@letterspace{\number\@tempcntb}%
1920     \fi
1921   }%
1922 }%

```

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

```

\MT@tr@ospace 1923 \MT@get@tr@opt@{spacing} {ispace}%
1924 \MT@get@tr@opt@{outerspacing}{ospace}%

```

`\MT@tr@okern` Adjust outer kerning.

```

1925 \MT@get@tr@opt@{outerkerning}{okern}%

```

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

```

1926 \MT@get@tr@opt@{noligatures} {ligatures}%
1927 }

```

`\MT@get@tr@opt@`

```

1928 \def\MT@get@tr@opt@#1#2{%
1929   \MT@ifdefined@nT{MT@tr@cc@MT@tr@cc@name @#1}{%
1930     {\MT@let@nn{MT@tr@#2}{MT@tr@cc@MT@tr@cc@name @#1}}%

```

```

1931 }
1932 </pdfTeX-def|luatex-def>
\MT@set@1sfont    Redefine \font@name, which will be called a second later (in \selectfont).
1933 < *pdfTeX-def|luatex-def|letterspace>
1934 <plain>\MT@requires@1atex2{
1935 \def\MT@set@1sfont{\MT@exp@two@c\let\font@name\MT@1sfont}

\1sstyle    Disable the tests whether the font should be letterspaced, then trigger the setup.
            Only \text1s can be used in math mode (\1sstyle may be used inside another
            text switch, of course). Still, we have to ensure that math fonts are set up again.
            Setting \g1b@currsz 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
            \g1b@settings instead.
1936 \DeclareRobustCommand\1sstyle{%
1937   \not@math@alphabet\1sstyle\text1s
1938 <pdfTeX-def|luatex-def> \MT@maybe@gobble@with@tikz{\aftergroup\g1b@settings}%
1939 <pdfTeX-def|luatex-def> \def\MT@feat{tr}%
1940 \let\MT@tracking\MT@set@tr@codes
1941 \selectfont
1942 }

            Now the definitions for the letterspace package with plain TEX.
1943 <*plain>
1944 }{
1945 \def\MT@set@1sfont{\MT@1sfont}
1946 \def\1sstyle{%
1947   \begingroup
1948   \escapechar\m@ne
1949   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1950   \MT@set@tr@codes
1951   \endgroup
1952 }
1953 \let\text1s\@undefined
1954 \let\1slig\@undefined
1955 }
1956 </plain>

\1slig    For Fraktur fonts, some ligatures shouldn't be broken up. This command will
            temporarily select the base font and insert the correct kerning.
1957 \DeclareRobustCommand\1slig[1]{%
1958   {\MT@ifdefined@c@TF\MT@curr@1s{%
1959     \escapechar\m@ne
1960     \MT@get@1s@basefont
1961     \MT@outer@kern=\dimexpr\MT@curr@1s sp * \fontdimen6\font@name/2000\relax
1962     \kern\MT@outer@kern
1963     \font@name #1%
1964     \kern\MT@outer@kern
1965   }}{#1}}%
1966 }

\MT@1s@basefont    pdfTEX cannot letterspace fonts that already are letterspaced. Therefore, we have
\MT@get@1s@basefont to save the base font in \<font name>@base.
            The previous solution (checking the macro's meaning with \pdfmatch), where
            we were loading the base font via the \font primitive again, would destroy all
            previously set up micro-typographic features of the font.
1967 \def\MT@get@1s@basefont{%
1968   \xdef\MT@1s@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1969   \expandafter\ifx\MT@1s@basefont\relax
1970     \MT@exp@two@c\MT@glet\MT@1s@basefont\font@name
1971   \else
1972 <debug>\MT@info@n1{1}{... fixing base font}%

```

```

1973 \MT@exp@two@c\let\font@name\MT@ls@basefont
1974 \fi
1975 }
\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.
1976 \def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
1977 \def\MT@set@tr@zero{%
1978 debug\MT@dinfnl{1}{... zero tracking}%
1979 \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1980 \expandafter\ifx\MT@ls@basefont\relax \else
1981 debug\MT@dinfnl{1}{... fixing base font}%
1982 \aftergroup\MT@set@lsbasefont
1983 \fi
1984 }
1985 pdfTeX-def|luatex-def|letterspace
\MT@tr@noligatures pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.
1986 pdfTeX-def|luatex-def
1987 pdfTeX-def\MT@requires@pdfTeX7{
1988 \def\MT@tr@noligatures{%
1989 \ifx\MT@tr@ligatures\empty
1990 \MT@noligatures@\MT@lsfont\undefined
1991 \else
1992 \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1993 \fi
1994 }
1995 pdfTeX-def
1996 }{
1997 \def\MT@tr@noligatures{%
1998 \MT@warning@n1{%
1999 Disabling selected ligatures is only possible since\MessageBreak
2000 pdfTeX 1.40.4. Disabling all ligatures instead}%
2001 \MT@glet\MT@tr@noligatures\relax
2002 }
2003 }
2004 pdfTeX-def
\MT@outer@space A new skip for outer spacing.
2005 \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.
2006 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
2007 debug\MT@dinfnl2{... orig. space: \the\fontdimen2\MT@lsfont,
2008 debug \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
2009 debug \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
2010 \let\MT@temp\empty
2011 \MT@tr@set@space@{#1}{#4}{2}\@empty
2012 \MT@tr@set@space@{#2}{#5}{3}\@plus
2013 \MT@tr@set@space@{#3}{#6}{4}\@minus
2014 \MT@glet@nc{\MT@outer@space\expandafter\string\font@name}\MT@temp
2015 debug\MT@dinfnl2{... inner space: \the\fontdimen2\MT@lsfont,
2016 debug \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
2017 debug\MT@dinfnl2{... outer space: \MT@temp}%
2018 }
\MT@tr@set@space@ If settings for outer spacing (#2) don't exist, they will be inherited from the inner
spacing settings (#1).
2019 \def\MT@tr@set@space@#1#2#3#4{%
2020 \MT@ifempty{#2}{%
2021 \MT@ifempty{#1}{%
2022 \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%

```

```

2023 }{%
2024   \MT@tr@set@space@{#1}{#3}{1000}%
2025   \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2026   \fontdimen#3\MT@lsfont=\@tempdima
2027 }%
2028 }{%
2029   \MT@tr@set@space@{#2}{#3}{2000}%
2030   \edef\MT@temp{\MT@temp#4\the\@tempdima}%
2031   \MT@ifempty{#1}\relax{%
2032     \MT@tr@set@space@{#1}{#3}{1000}%
2033     \fontdimen#3\MT@lsfont=\@tempdima
2034   }%
2035 }%
2036 }

```

`\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.

```

2037 \def\MT@tr@set@space@#1#2#3{%
2038   \MT@test@ast#1*\nil{%
2039     \MT@ifdefined@cTF\MT@tr@unit@
2040     {\edef\@tempb{#1}\MT@scale@to@em}
2041     {\@tempcntb=#1\relax}%
2042     \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
2043     -\fontdimen#2\MT@lsfont\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).

```

2044   \ifnum#2=\tw@
2045     \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
2046   \fi
2047   \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
2048 }{%
2049   \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
2050   \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
2051 }%
2052 (debug)\MT@dinfo@n13{... : font dimen #2 (#1): \the\@tempdima}%
2053 }

```

`\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).

```

2054 \def\MT@tr@outer@l{%
2055   \ifhmode
2056     \ifdim\lastskip>5sp
2057       \edef\x{\the\lastskip minus 0pt}%
2058       \setbox\z@\hbox{\MT@outer@space=\x}%
2059       \ifdim\wd\z@>\z@
2060 (debug)\MT@dinfo@2{[[[ adjusting pre space: \the\MT@outer@space}%
2061         \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

2062   \let\MT@ls@outer@k\relax
2063   \else

```

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

```

2064   \ifdim\lastskip=%
2065     \ifnum\spacefactor<2000
2066       \spaceskip
2067     \else
2068       \ifdim\xspaceskip=\z@
2069         \dimexpr\spaceskip+\fontdimen7\font@name\relax
2070       \else
2071         \xspaceskip
2072       \fi
2073     \fi

```

```

2074 (debug)\MT@dinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
2075         \unskip \hskip\MT@outer@space\relax
2076         \let\MT@ls@outer@k\relax
2077         \fi
2078     \fi
2079     \fi
2080     \fi
2081 }

```

`\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`).

```

2082 \def\MT@tr@outer@r{%
2083     \futurelet\MT@tr@outer@next\MT@tr@outer@r@
2084 }

```

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

```

2085 \def\MT@if@outer@next#1{%
2086     \ifx\MT@tr@outer@next#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
2087 }

```

`\MT@tr@outer@r@`

```

2088 \def\MT@tr@outer@r@{%
2089     \def\MT@temp*{}%

```

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

```

2090     \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).

```

2091     \ifnum\currentgrouptype=10 \else
2092         \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
2093 (debug)\MT@dinfo2{[[[ adjusting post space (1): \the\MT@outer@space}%
2094         \fi}%
2095         \expandafter\ifcat\expandafter\noexpand\cname \MT@tr@outer@next\endcname\egroup
2096         \ifhmode\unkern\fi\egroup
2097         \MT@set@curr@ok \MT@set@curr@os
2098         \def\MT@temp*{\afterassignment\MT@tr@outer@r@let\MT@temp=%}%
2099     \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.

```

2100     \MT@if@outer@next\maybe@ic{%
2101         \MT@set@curr@ok \MT@set@curr@os
2102         \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=%}%
2103     }{%

```

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).

```

2104     \MT@if@outer@next\check@icr{%
2105         \def\MT@temp*{\aftergroup\MT@tr@outer@r@check@icr\let\MT@temp=%}%
2106     }{%
2107         \MT@if@outer@next\@sptoken{%
2108             \def\MT@temp* {\ifhmode\hskip\MT@outer@space
2109 (debug)\MT@dinfo2{[[[ adjusting post space (2): \the\MT@outer@space}%
2110             \fi}%
2111         }{%

```

```

2112             \MT@if@outer@next~{%
2113             \def\MT@temp*~{\nobreak\hskip\MT@outer@space
2114 (debug)\MT@dinfo2{[]]} adjusting post space (3): \the\MT@outer@space}%
2115             }%
2116         }{%
2117             \MT@if@outer@next\ \relax{%
2118             \MT@if@outer@next\space\relax{%
2119             \MT@if@outer@next\@xobeysp\relax{%

```

xspace requires special treatment.

```

2120             \MT@if@outer@next\xspace{%
2121             \def\MT@temp*\xspace{\MT\xspace}%
2122             }{%

```

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

```

2123             \def\MT@temp*{\ifdim\MT@outer@kern=\z@else\MT@ls@outer@k
2124 (debug)\MT@dinfo2{--- adjusting post kern: \the\MT@outer@kern}%
2125             \fi}%
2126             \MT@let@nc{\MT@tr@outer@next}\relax
2127         }}}}]]}}\fi
2128     \fi\fi
2129     \MT@temp*%
2130 }

```

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

```

\MT@tr@outer@icr@ 2131 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
2132 \def\MT@tr@outer@icr@{%
2133     \let\@let@token= \MT@tr@outer@next
2134     \maybe@ic@
2135 }

```

`\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.

```

2136 \def\MT@xspace{\futurelet\@let@token\MT@xspace@}
2137 \def\MT@xspace@{\@xspace@firsttrue\xspace
2138     \ifdim\lastskip>5sp
2139     \unskip \hskip\MT@outer@space
2140     \else
2141     \ifdim\MT@outer@kern=\z@else\MT@ls@outer@k \fi
2142     \fi
2143 }

```

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

```

2144 }{
2145     \DeclareRobustCommand\lsstyle{%
2146         \MT@error{Letterspacing only works with \MT@engine tex version
2147 (pdfTEX-def) 1.40%
2148 (luaTEX-def) 0.62%
2149         \MessageBreak or newer}
2150         {Upgrade \MT@engine tex, or try the `soul' package instead.}%
2151         \MT@glet\lsstyle\relax
2152     }
2153 }

```

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

```

2154 (/pdfTEX-def|luaTEX-def)
2155 (*xTEX-def)
2156 \DeclareRobustCommand\lsstyle{%
2157     \MT@error{Letterspacing currently doesn't work with xetex}
2158     {Run pdfTEX or luaTEX, or use the `soul' package instead.}%
2159     \MT@glet\lsstyle\relax
2160 }
2161 (/xTEX-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.
2162 (*package|letterspace)
2163 \DeclareRobustCommand\textls{%
2164   \ifstar{\let\MT@ls@adjust@MT@ls@adjust@empty\MT@textls}%
2165         {\let\MT@ls@adjust@MT@ls@adjust@relax\MT@textls}%
2166 }

\MT@textls   This is now almost LATEX's \DeclareTextFontCommand, with the difference that we
\MT@letterspace adjust the outer spacing and kerning also for \lssstyle, while LATEX's text switches
              don't bother about italic correction.
2167 \newcommand\MT@textls[2][ ]{%
2168   \ifmode
2169     \nfss@text{\MT@ls@set@ls{#1}\lssstyle#2}%
2170   \else
2171     \hmode@bgroup
2172     \MT@ls@set@ls{#1}%
2173     \lssstyle #2%
2174     \expandafter
2175     \egroup
2176   \fi
2177 }

\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 2178 \def\MT@ls@adjust@empty{\let\MT@ls@adjust@empty}
\MT@ls@set@ls      2179 \def\MT@ls@adjust@relax{\let\MT@ls@adjust@relax}
2180 \def\MT@ls@set@ls#1{%
2181   \MT@ifempty{#1}%
2182   {\let\MT@letterspace@\undefined}%
2183   {\KV@sp@def\MT@letterspace@{#1}%
2184     \edef\MT@letterspace@{\number\MT@letterspace@}%
2185     \MT@ls@too@large\MT@letterspace@}%
2186   \MT@ls@adjust@
2187 }

\MT@ls@too@large   Test whether letterspacing amount is too large.
2188 \def\MT@ls@too@large#1{%
2189   \ifnum#1>\MT@tr@max
2190     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
2191     \let#1\MT@tr@max
2192   \else
2193     \ifnum#1<\MT@tr@min
2194       \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
2195       \let#1\MT@tr@min
2196     \fi
2197   \fi
2198 }

\MT@outer@kern     This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern   outer kerning.
2199 \newdimen\MT@outer@kern
2200 (/package|letterspace)
2201 (*pdftex-def|luatex-def)
2202 \def\MT@tr@set@okern#1,#2,{%
2203   \let\MT@temp@empty
2204   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
2205   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
2206   \MT@glet@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
2207 (debug)\MT@dinfo@n12{... outer kerning: (#1,#2)
2208 (debug)           = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%
2209 }

```

`\MT@tr@set@okern@`

```

2210 \def\MT@tr@set@okern@#1{%
2211   \MT@test@ast#1*\@nil{%
2212     \MT@ifdefined@c@TF\MT@tr@unit@
2213       {\edef\@tempb{#1}\MT@scale@to@em}
2214       {\@tempcntb=#1\relax}%
2215     \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
2216   }{%
2217     \MT@ifempty\@tempa{\let\@tempa\@m}\relax
2218     \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
2219       * \fontdimen6\MT@lsfont/2000\relax
2220   }%
2221   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
2222       * \fontdimen6\MT@lsfont/2000\relax
2223   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
2224 }
2225 </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.

```

2226 <*pdfTeX-def|luatex-def|letterspace>
2227 \def\MT@ls@outer@k{%
2228   \ifhmode
2229     \ifdim\lastkern=-3sp \unkern
2230     \ifdim\lastkern=3sp \kern-3sp
2231     \expandafter\expandafter\expandafter\@gobble
2232     \else \unkern
2233     \expandafter\expandafter\expandafter\@firstofone
2234     \fi
2235   \else
2236     \expandafter\@firstofone
2237     \fi
2238   {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
2239   \fi
2240 }
2241 </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.

```

2242 <*pdfTeX-def|luatex-def>
2243 <pdfTeX-def>\MT@requires@pdfTeX5{
2244 \def\MT@noligatures{%
2245   \MT@dotrue
2246   \let\@tempa\MT@n@l@setname
2247   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
2248     \MT@ifdefined@n@TF{MT@checklist@##1}%
2249     {\csname MT@checklist@##1\endcsname}%
2250     {\MT@checklist@{##1}}%
2251   {n1}}%
2252 }%
2253 \ifMT@do
2254   \MT@noligatures@MT@font\MT@n@l@ligatures
2255   \fi
2256 }

```

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

```

2257 <luatex-def>\MT@requires@luatex4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
2258 \def\MT@noligatures@#1#2{%
2259   \MT@ifdefined@c@TF#2{%

```

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

```
2260 \MT@ifdefined@c@TF\tagcode{%
```

No ‘inputenc’ key.

```
2261 \let\MT@warn@maybe@inputenc\@empty
2262 \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
2263 \MT@map@clist@c#2{%
2264 \KV@sp@def\@tempa{#1}\MT@get@slot
2265 \ifnum\MT@char>\m@ne
2266 \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`).

```
2267 <luatex-def> \MT@if@fontspec@font
2268 <luatex-def> {\MT@lua{microtype.noligatures([[#1]],[[\MT@char]])}}\relax
2269 \fi
2270 }%
2271 \MT@vinfo{... Disabling ligatures for characters: #2}%
2272 }{%
2273 \pdfnoligatures#1%
2274 \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
2275 know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
2276 the font instead}%
2277 }%
2278 }{%
2279 \pdfnoligatures#1%
2280 <luatex-def> \MT@if@fontspec@font
2281 <luatex-def> {\MT@lua{microtype.noligatures([[#1]],"_all_")}}\relax
2282 \MT@vinfo{... Disabling all ligatures}%
2283 }%
2284 }
2285 <pdftex-def>}\relax
2286 </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@`.

```
2287 <*luafile>
2288 microtype.ligs = microtype.ligs or { }
2289
2290 local function noligatures(fontcs,liga)
2291 local fontcs = match(fontcs,"([^\ ]+)"
2292 microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
2293 table.insert(microtype.ligs[fontcs],liga)
2294 end
2295 microtype.noligatures = noligatures
2296
2297 local function keepligature(c)
2298 local nodedirect = node.direct
2299 local getfield = nodedirect.getfield
2300 local getfont = nodedirect.getfont
2301 local f,ch
2302 if type(c) == "userdata" then -- in older luaotfload versions, c was a node
2303 f = c.font
2304 ch = c.components.char
2305 else -- since 2.6, c is a (direct node) number
2306 f = getfont(c)
2307 ch = getfield(getfield(c,"components"),"char")
2308 end
2309 -- if ch then -- should always be true
2310 local ligs = microtype.ligs[match(tex.fontidentifier(f),"\\([^\ ]+)")
2311 if ligs then
2312 for _,lig in pairs(ligs) do
2313 if lig == "_all_" or tonumber(lig) == ch then
```

```

2314     return false
2315   end
2316 end
2317 end
2318 return true
2319 -- end
2320 end
2321
2322 if luaotfload and luaotfload.letterspace then
2323   if luaotfload.letterspace.keepligature then
2324     microtype.warning("overwriting function `keepligature'")
2325   end
2326   luaotfload.letterspace.keepligature = keepligature
2327 end
2328
2329 /luafile

```

### 14.2.7 Loading the configuration

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

```

2330 <package>
2331 \def\MT@load@list#1{%
2332   \edef\@tempa{#1}%
2333   \MT@let@cn\@tempb{MT@MT@feat @c@\@tempa @load}%
2334   \MT@ifstreq\@tempa\@tempb{%
2335     \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempa' cannot load itself}{}%
2336   }{%
2337     \ifx\@tempb\relax \else
2338       \MT@ifdefined@n@TF{MT@MT@feat @c@\@tempb}{%
2339         \MT@vinfo{... : First loading \@nameuse{MT@abbr@MT@feat} list `@\@tempb'}%
2340         \begin@group
2341           \MT@load@list\@tempb
2342         \end@group
2343         \edef\MT@curr@list@name{\@nameuse{MT@abbr@MT@feat} list
2344           \noexpand\MessageBreak`@\@tempb'}%
2345         \MT@let@cn\@tempc{MT@MT@feat @c@\@tempb}%
2346         \expand@after\MT@set@codes\@tempc,\relax,%
2347       }{%
2348         \MT@error{\@nameuse{MT@abbr@MT@feat} list `@\@tempb' undefined.\MessageBreak
2349           Cannot load it from list `@\@tempa'}{}}%
2350     }%
2351   \fi
2352 }%
2353 }

```

`\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.

```

2354 \let\MT@file@list\@empty
2355 \def\MT@find@file#1{%

```

Check for existence of the file only once.

```

2356   \MT@in@clist{#1}\MT@file@list
2357   \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.

```

2358   \MT@begin@catcodes
2359   \let\MT@begin@catcodes\relax
2360   \let\MT@end@catcodes\relax
2361   \InputIfFileExists{mt-#1.cfg}{%
2362     \edef\MT@curr@file{mt-#1.cfg}%
2363     \MT@vinfo{... Loading configuration file \MT@curr@file}%
2364     \MT@xadd\MT@file@list{#1,%

```

```

2365 }{%
2366 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
2367 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2368 \ifMT@inlist@
2369 \MT@xadd\MT@file@list{#1,}%
2370 \else
2371 \InputIfFileExists{mt-\@tempa.cfg}{%
2372 \edef\MT@curr@file{mt-\@tempa.cfg}%
2373 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2374 \MT@xadd\MT@file@list{\@tempa,#1,}%
2375 }{%
2376 \MT@vinfo{... No configuration file mt-#1.cfg}%
2377 \MT@xadd\MT@file@list{#1,}%
2378 }%
2379 \fi
2380 }%
2381 \endgroup
2382 \fi
2383 }

```

`\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.

```

2384 \def\MT@cfg@catcodes{%
2385 \makeatletter
2386 \catcode`\^7%
2387 \catcode`\ 9%
2388 \catcode`\^^I9%
2389 \catcode`\^^M9%
2390 \catcode`\\z@
2391 \catcode`\{\@ne
2392 \catcode`\}\@tw@
2393 \catcode`\#6%
2394 \catcode`\%14%
2395 \MT@map@tlist@n
2396 {\!\"$&\'(\)\|*+,\|-\.\/\:\;\<=\>|\?[\]\_-\|\/}%
2397 \makeoother
2398 }

```

`\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.

```

2399 \def\MT@begin@catcodes{%
2400 \begin@group
2401 \MT@cfg@catcodes
2402 }

```

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

```

2403 \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.

```

2404 \def\MT@get@basefamily#1#2#3#4\@nil{%
2405 \ifx\@empty#4%
2406 \def\@tempa{#1#2#3}%

```

Table 4:

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

```

2407 \else
2408   \let\@tempa\@empty
2409   \edef\@tempb{#1#2#3#4}%
2410   \expandafter\MT@get@basefamily@\@tempb\@nil
2411 \fi
2412 }

```

`\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 ‘padx’ would be truncated to ‘p’.

```

2413 \def\MT@get@basefamily@#1#2\@nil{%
2414   \edef\@tempa{\@tempa#1}%
2415   \ifx\#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
2416   {\MT@in@tlist{#2}\MT@variants
2417    \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
2418 }

```

`\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@ 2419 \def\MT@get@listname#1{%
2420   (debug)\MT@info@n1{1}{trying to find \nameuse{MT@abbr@#1} list for font ` \MT@font'}%
2421   \let\MT@listname\undefined
2422   \def\@tempb{#1}%
2423   \MT@map@tlist@c\MT@try@order\MT@get@listname@
2424 }
2425 \def\MT@get@listname@#1{%
2426   \expandafter\MT@next@listname#1%
2427   \ifx\MT@listname\undefined \else
2428     \expandafter\MT@tlist@break
2429   \fi
2430 }

```

`\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.

```

2431 \def\MT@try@order{%
2432   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2433   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2434 }

```

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

```

2435 \def\MT@next@listname#1#2#3#4{%
2436   \ifnum#1=\z@\MT@nofamilytrue\fi
2437   \edef\@tempa{\MT@encoding
2438    /\ifnum#1=\@ne \MT@family \fi
2439    /\ifnum#2=\@ne \MT@series \fi
2440    /\ifnum#3=\@ne \MT@shape \fi
2441    /\ifnum#4=\@ne *\fi
2442    \MT@context}%
2443   (debug)\MT@info@n1{1}{trying \@tempa}%
2444   \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}%
2445   \MT@next@listname@#4%

```

```
2446 }%
```

Also try with an alias family.

```
2447 \ifnum#1=\@ne
2448 \ifx\MT@familyalias\@empty \else
2449 \edef\@tempa{\MT@encoding
2450 /MT@familyalias
2451 /\ifnum#2=\@ne \MT@series\fi
2452 /\ifnum#3=\@ne \MT@shape\fi
2453 /\ifnum#4=\@ne *\fi
2454 \MT@context}%
2455 (debug)\MT@info@n1{1}{(alias) \@tempa}%
2456 \MT@ifdefined@nT{MT@\@tempb @\@tempa}{%
2457 \MT@next@listname@#4%
2458 }%
2459 \fi
2460 \fi
2461 }%
2462 }
```

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

```
2463 \def\MT@next@listname@#1{%
2464 \ifnum#1=\@ne
2465 \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
2466 \ifMT@inlist@
2467 \let\MT@listname\MT@size@name
2468 \fi
2469 \else
2470 \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2471 \fi
2472 }
```

\MT@if@list@exists

```
\MT@context 2473 \def\MT@if@list@exists{%
2474 \MT@let@cn\MT@context{MT@\MT@feat @context}%
2475 \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2476 \MT@get@listname{\MT@feat @c}%
2477 \MT@ifdefined@c@TF\MT@listname{%
2478 \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2479 \ifMT@nonselected
2480 \MT@vinfo{... Applying non-selected expansion (list `MT@listname')}%
2481 \else
2482 \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `MT@listname'}%
2483 \fi
2484 \@firstoftwo
2485 }%
```

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

```
2486 \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```
2487 \ifMT@nonselected
2488 \MT@vinfo{... Applying non-selected expansion (no list)}%
2489 \else
```

Tracking doesn't require a list, either.

```
2490 \MT@ifstreq\MT@feat{tr}\relax%
2491 \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2492 for font\MessageBreak`MT@font'%
2493 \ifx\MT@context\@empty\else\space(context: `MT@context')\fi.
2494 Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
2495 }%
2496 \fi
2497 \@secondoftwo
```

```

2498 }%
2499 }
\MT@get@inh@list    The inheritance lists are global (no context).
\MT@context 2500 \def\MT@get@inh@list{%
2501   \let\MT@context\@empty
2502   \MT@get@listname{\MT@feat @inh}%
2503   \MT@ifdefined@c@TF\MT@listname{%
2504     \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
2505     (debug)\MT@dinfo@n1{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
2506     (debug)         ~\MT@listname'}%
2507     \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%

```

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

```

2508   \ifx\@tempc\@empty \else
2509     (debug)\MT@dinfo@n1{1}{parsing inheritance list ...}%

```

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

```

2510     \begingroup
2511     \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak~\MT@listname'}%
2512     \MT@set@inputenc{inh}%
2513     \expandafter\MT@inh@do\@tempc,\relax,%
2514     \MT@gl@et@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2515     \endgroup
2516   \fi
2517 }{%
2518   \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2519 }%
2520 }

```

### 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@ 2521 \def\MT@get@slot{%
2522   \escapechar~\
2523   \let\MT@char@m@ne
2524   \MT@noesttrue

```

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

```

2525   \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.

```

2526   \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.

```

2527   \expandafter\MT@is@letter\@tempa\relax\relax
2528   \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  $\backslash i$  or  $\backslash U\backslash CYRI$ , hence,  $\backslash string$  wouldn't be safe enough.

```
2529 \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c@tempa}%
2530 \MT@is@symbol
```

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

```
2531 {\expandafter\MT@is@composite@tempa\relax\relax}%
2532 \ifnum\MT@char@ < \z@
```

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

```
2533 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2534 \meaning\expandafter\tempa\MT@charstring\relax\relax\relax
2535 \fi
2536 \fi

2537 \let\MT@char\MT@char@
2538 \MT@get@slot@
2539 \escapechar\m@ne
2540 }
2541 /package
```

$\backslash MT@get@slot@$

```
2542 <pdfTeX-def|luatex-def|xetex-def
2543 \def\MT@get@slot@{%
```

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

```
2544 <xetex-def \ifnum\XeTeXfonttype\MT@font=\z@
2545 \ifnum\MT@char > \m@ne
```

In LuaTeX, it may also be a glyph name, prefixed with  $\backslash$ .

```
2546 <luatex-def
2547 \ifnum\MT@char=47\relax
2548 \ifMT@noest \else
2549 \@tempcnta=\MT@lua{
2550 local glyph = microtype.name_to_slot([[expandafter\gobble\tempa]],true)
2551 if glyph then tex.write(glyph)
2552 else tex.write(-1)
2553 end
2554 }\relax
2555 \ifnum\@tempcnta<\z@
2556 \MT@warn@unknown
2557 \let\MT@char\m@ne
2558 \else
2559 \edef\MT@char{\the\@tempcnta}%
2560 <debug\MT@info@n1{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
2561 \fi
2562 \fi
2563 \else
2564 /luatex-def
```

If the user has specified something like  $\backslash fi$ , 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.

```
2565 \ifMT@noest \else
2566 \MT@warn@rest
2567 <pdfTeX-def|luatex-def \let\MT@char\m@ne
2568 <xetex-def \let\MT@char\@empty
```

```

2569   \fi
2570 (luatex-def)   \fi
2571   \else
2572     \MT@warn@unknown
2573 (xetex-def)   \let\MT@char\@empty
2574   \fi
2575 (*xetex-def)
2576   \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@.

```

2577   \ifnum\MT@char=47\relax
2578     \ifMT@noreset \edef\MT@char{U47}%
2579   \else
2580     \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
2581     \ifnum\@tempcnta=\z@
2582       \MT@warn@unknown
2583       \let\MT@char\@empty
2584     \else
2585       \edef\MT@char{\@tempa\space}%
2586       \edef\MT@char@{-\the\@tempcnta}%
2587 (debug)\MT@debug@n1{3}{> `the\MT@toks' is a glyph name (\the\@tempcnta)}%
2588     \fi
2589   \fi
2590   \else
2591     \ifnum\MT@char > \m@ne
2592     \ifMT@noreset

```

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

```

2593     \@tempcnta=\XeTeXcharglyph\MT@char\relax
2594     \ifnum\@tempcnta=\z@
2595       \MT@info@missing@char
2596       \let\MT@char\@empty
2597     \else
2598 (debug)\MT@debug@n1{3}{> (glyph number: \the\@tempcnta,
2599 (debug)      glyph name: \XeTeXglyphname\MT@font\@tempcnta)}%
2600     \edef\MT@char{U\MT@char}%
2601     \fi
2602   \else
2603     \MT@warn@rest
2604     \let\MT@char\@empty
2605   \fi
2606   \else
2607     \MT@warn@unknown
2608     \let\MT@char\@empty
2609   \fi
2610   \fi
2611 (/xetex-def)
2612 }
2613 }
2614 (/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.

```

2615 (*luafile)
2616 if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
2617   local slot_of_name = luaotfload.aux.slot_of_name
2618   microtype.name_to_slot = function(name, unsafe)
2619     return slot_of_name(font.current(), name, unsafe)
2620   end
2621 else
2622   -- we dig into internal structure (should be avoided)

```

```

2623 local function name_to_slot(name, unsafe)
2624   if fonts then
2625     local unicodes
2626     if fonts.ids then --- legacy luaotfload
2627       local tfmdata = fonts.ids[font.current()]
2628       if not tfmdata then return end
2629       unicodes = tfmdata.shared.otfdata.luatex.unicodes
2630     else --- new location
2631       local tfmdata = fonts.hashes.identifiers[font.current()]
2632       if not tfmdata then return end
2633       unicodes = tfmdata.resources.unicodes
2634     end
2635     local unicode = unicodes[name]
2636     if unicode then --- does the 'or' branch actually exist?
2637       return type(unicode) == "number" and unicode or unicode[1]
2638     end
2639   end
2640 end
2641 microtype.name_to_slot = name_to_slot
2642 end
2643
2644 (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 2645 (*pdfTeX-def|luatex-def|xetex-def)
2646 \def\MT@max@char
2647 (pdfTeX-def) {127 }
2648 (luatex-def|xetex-def) {1114111 }
2649 \def\MT@max@slot
2650 (pdfTeX-def) {255 }
2651 (luatex-def|xetex-def) {1114111 }
2652 (pdfTeX-def|luatex-def|xetex-def)

```

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

```

2653 (*package)
2654 \newif\ifMT@noest

2655 \def\MT@is@letter#1#2\relax{%
2656   \ifcat a\noexpand#1\relax
2657     \edef\MT@char@{\number`#1}%
2658     \ifx\#2\%
2659 (debug)\MT@info@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2660     \else
2661       \MT@noestfalse
2662     \fi
2663   \else
2664     \ifcat !\noexpand#1\relax
2665       \edef\MT@char@{\number`#1}%
2666 (debug)\MT@info@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
2667       \ifx\#2\%
2668         \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
2669       \else
2670         \MT@noestfalse
2671         \expandafter\MT@is@number#1#2\relax\relax
2672       \fi
2673     \fi
2674   \fi
2675 }

```

`\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.

```

2676 \def\MT@is@number#1#2#3\relax{%
2677   \ifx\relax#3\relax \else
2678     \ifx\relax#2\relax \else
2679       \MT@noesttrue
2680       \if#1"\relax
2681         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2682 (debug)\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2683       \else
2684         \if#1'\relax
2685           \def\MT@char@{\number#1#2#3}%
2686 (debug)\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2687         \else
2688           \MT@ifint{#1#2#3}{%
2689             \def\MT@char@{\number#1#2#3}%
2690 (debug)\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2691           }\MT@noestfalse
2692         \fi
2693       \fi
2694       \ifnum\MT@char@ > \MT@max@slot
2695         \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2696         \let\MT@char@\m@ne
2697       \fi
2698     \fi
2699   \fi
2700 }

```

`\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.

```

2701 \def\MT@is@active#1#2\@nil{%
2702   \ifnum\catcode`#1 = \active
2703     \begingroup
2704     \set@display@protect
2705     \let\IeC\@firstofone
2706     \let\@inpenc@undefined@\MT@undefined@char

```

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

```

2707   \def\UTFviii@defined##1{\ifx ##1\relax
2708     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%

```

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

```

2709   \MT@ifdefined@c@T\PrerenderUnicode
2710     {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%

```

The `\expandafter` hocus-pocus should please `newunicodechar`.

```

2711   \edef\x{\endgroup
2712     \def\noexpand\@tempa{\expandafter\expandafter\expandafter\@empty\@tempa}%

```

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

```

2713     \MT@toks={\the\MT@toks\space(=
2714               \expandafter\expandafter\expandafter\@empty\@tempa)}%
2715   }%
2716   \x
2717   \fi
2718 }

```

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

```

2719 \def\MT@undefined@char#1{undefined in input encoding ``#1''}

```

`\MT@is@symbol` The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding `\langle command \rangle`, we construct the command `\langle encoding \rangle \langle command \rangle` and see whether its meaning is `\char " \langle hex number \rangle`, which is the case for everything that has been defined with `\DeclareTextSymbol` in the encoding definition files.

```

2720 \def\MT@is@symbol{%
2721   \expandafter\def\expandafter\MT@char\expandafter
2722     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2723   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2724     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2725   \ifnum\MT@char@ < \z@

```

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

```

2726     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2727     \fi
2728 }

```

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

`\MT@charstring`

```

2729 \begingroup
2730   \catcode`\=/\z@

```

```

2731   /MT@map@tlist@n{/CHARLEX}/@makeoether
2732   /lowercase{%
2733     /def/x{/endgroup
2734       /def/MT@charstring{\CHAR"%}
2735       /def/MT@is@char##1\CHAR"##2##3##4/relax{%
2736         /ifx/relax##4/relax
2737         /ifMT@xunicode
2738           /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
2739           /relax/relax/relax/relax/relax
2740         /fi
2741       /else
2742         /ifx/relax##1/relax
2743         /if##3\relax
2744         /edef/MT@char@{/number"##2}%
2745         /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
2746       /else
2747         /edef/MT@char@{/number"##2##3}%
2748         /MT@ifstreq/MT@charstring{##4}/relax
2749         {/MT@is@xchar##2##3|##4\CHAR"/relax}%
2750       /fi
2751   <debug> /MT@dinfo@n1{3}{> `~/the/MT@toks' is a \char (/MT@char@)}%
2752   /fi
2753 /fi
2754 }%

```

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

```

2755   /def/MT@is@xchar##1|##2\CHAR"##3##4/relax{%
2756     /MT@ifstreq/MT@charstring{##3##4}%
2757     {/edef/MT@char@{/number"##1##2}}/MT@noestfalse
2758   }%

```

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

`\MT@strip@prefix`

```

2759   /def/MT@charxstring{\CHAR "%}

```

`\MT@is@charx`

```

2760   /def/MT@strip@prefix##1>##2/relax{##2}%
2761   /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
2762     /ifx/relax##1/relax
2763     /ifx/relax##6/relax/else
2764       /edef/MT@char@{/number"##2##3##4##5}%
2765       /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@noestfalse
2766   <debug> /MT@dinfo@n1{3}{> `~/the/MT@toks' is a xunicode \char (/MT@char@)}%

```

```

2767         /fi
2768     /fi
2769     }%
2770 }%
2771 }
2772 /x

```

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

```

2773 \def\MT@is@composite#1#2\relax{%
2774     \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`.

```

2775     \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2776         \string\csname\MT@encoding\endcsname
2777         \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):

```

2778     \ifx\UnicodeEncodingName\undefined\else
2779     \expandafter\expandafter\expandafter
2780     \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
2781     \fi
2782     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax

```

Again, `xunicode`.

```

2783     \ifnum\MT@char@ < \z@
2784     \ifMT@xunicode
2785     \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
2786     \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
2787     \MT@char\MT@charxstring\relax\relax\relax\relax\relax
2788     \fi
2789     \fi
2790     \fi
2791 }

```

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

```

2792 \def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
2793     \ifx\#2\\\else\edef\MT@char{\iffontchar#2\fi}\fi
2794 }

```

[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 2795 \def\MT@set@listname{%
2796   \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2797   \@nameuse{MT@\MT@feat @c@name}}}%
2798 }

\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.
2799 \def\MT@warn@ascii{%
2800   \MT@warning@nl{Character `the\MT@toks' (= \MT@char@)
2801   is outside of ASCII range.\MessageBreak
2802   You must load the `inputenc' package before using\MessageBreak
2803   8-bit characters in \MT@curr@list@name}%
2804 }

\MT@warn@number@too@large    Number too large.
2805 \def\MT@warn@number@too@large#1{%
2806   \MT@warning@nl{%
2807     Number #1 in encoding ` \MT@encoding' too large!\MessageBreak
2808     Ignoring it in \MT@curr@list@name}%
2809 }

\MT@warn@rest    Not all of the string has been parsed.
2810 \def\MT@warn@rest{%
2811   \MT@warning@nl{%
2812     Unknown slot number of character\MessageBreak`the\MT@toks'%
2813     \MT@warn@maybe@inputenc\MessageBreak
2814     in font encoding ` \MT@encoding'.\MessageBreak
2815     Make sure it's a single character\MessageBreak
2816     (or a number) in \MT@curr@list@name}%
2817 }

\MT@warn@unknown    No idea what went wrong.
2818 \def\MT@warn@unknown{%
2819   \MT@warning@nl{%
2820     Unknown slot number of character\MessageBreak`the\MT@toks'%
2821     \MT@warn@maybe@inputenc\MessageBreak
2822     in font encoding ` \MT@encoding' in \MT@curr@list@name}%
2823 }

\MT@warn@maybe@inputenc    In case an input encoding had been requested.
2824 \def\MT@warn@maybe@inputenc{%
2825   \MT@ifdefined@n@T
2826   {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}%
2827   { (input encoding ` \@nameuse
2828   {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2829 }

```

### 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 `pdfcpot` package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those

fonts that are actually being used will be set up.

For my reference:

- `\pickup@font` is called by `\selectfont`, `\wrong@fontshape`, or `\getanddefine@fonts` (for math).
- `\pickup@font` calls `\define@newfont`.
- `\define@newfont` may call (inside a group!)
  - `\wrong@fontshape`, which in turn will call `\pickup@font`, and thus `\define@newfont` again, or
  - `\extract@font`.
- `\get@external@font` is called by `\extract@font`, by itself, and by the substitution macros.

Up to version 1.3 of this package, we were using `\define@newfont` as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before `microtype` and were loading fonts, e.g., `jurabib`, `ledmac`, `pi font` (loaded by `hyperref`), `tipa`, and probably many more. Furthermore, we had to include a hack for the `IEEEtran` class which loads all fonts in the class file itself (to fine tune inter-word spacing), and the `memoir` class, too. To cut this short: it seemed to get out of hand, and I decided that it would be better to use `\pickup@font` and decide for ourselves whether we've already seen that font. I hope the overhead isn't too large.

`\MT@font@list` We use a comma separated list.

```
\MT@font 2830 \let\MT@font@list\@empty
2831 \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`.

```
2832 </package>
2833 <*package|letterspace>
2834 <plain>\MT@requires@1atex2{
2835 \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.

```
2836 <package> \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
2837 <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`.

```
2838 \@ifpackageloaded{CJK}{%
```

The xeCJK package in turn pretends that CJK was loaded, but does not change the definition of `\pickup@font`. With xeCJK, protrusion should be possible also for C/J/K characters; I haven't tried it, though.

```
2839 \ifpackageloaded{xeCJK}{\@firstofone}{%
2840 \ifpackagelater{CJK}{2006/10/17}% 4.7.0
2841 {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
2842 {\def\MT@orig@pickupfont{\ifundefined{CJK@plane}}}%
2843 \g@addto@macro\MT@orig@pickupfont
2844 {\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).

```
2845 \ifpackageloaded{CJKutf8}%
2846 {\ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
2847 {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
2848 {\@firstoftwo}}%
2849 {\@firstoftwo}%
2850 {\g@addto@macro\MT@orig@pickupfont{%
2851 \expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2852 \define@newfont\else\xdef\font@name{%
2853 \csname\curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}%
2854 {\g@addto@macro\MT@orig@pickupfont{%
2855 \expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2856 \define@newfont\def\CJK@temp{v}%
2857 \ifx\CJK@temp\CJK@plane
2858 \expandafter\ifx\csname CJK@cmapp/\f@family\CJK@plane\endcsname\relax
2859 \else\csname CJK@cmapp/\f@family\CJK@plane\endcsname\fi
2860 \else\CJK@addcmap\CJK@plane\fi
2861 \else\xdef\font@name{%
2862 \csname\curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}%
2863 \@gobble
2864 }%
2865 }{\@firstofone}}%
```

This is the normal L<sup>A</sup>T<sub>E</sub>X definition.

```
2866 {\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.

```
2867 \ifx\pickup@font\MT@orig@pickupfont\else
2868 \MT@warning@nl{%
2869 Command \string\pickup@font\space is not defined as expected.%
2870 \MessageBreak Patching it anyway. Some things may break%
2871 (*package)
2872 .\MessageBreak Double-check whether micro-typography is indeed%
2873 \MessageBreak applied to the document.%
2874 \MessageBreak (Hint: Turn on `verbose' mode)%
2875 (/package)
2876 }%
2877 \fi
```

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

```
2878 \g@addto@macro\pickup@font{\begingroup}%
```

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

```
2879 \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
2880 \g@addto@macro\pickup@font{%
2881 \escapechar\m@ne
2882 (*package)
2883 (debug) \global\MT@inannottrue
2884 (debug) \MT@gl@et\MT@pdf@annot\@empty
2885 (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.

```

2886     \MT@let@cn\MT@font{MT@subst@expandafter\string\font@name}%
2887     \ifx\MT@font\relax
2888       \let\MT@font\font@name
2889     \else
2890       \ifx\MT@font\font@name \else
2891 (debug) \MT@addto@annot{= substituted with \MT@font}%
2892         \MT@register@subst@font
2893       \fi
2894     \fi
2895     \MT@setupfont
2896 (/package)
2897 (letterspace)     \MT@tracking
2898     \endgroup
2899   }%
2900 (*package)

```

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

```

\MT@MT@pickupfont
\MT@ltx@pickupfont 2901 \let\MT@pickupfont\pickup@font
2902 \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
2903 \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.

```

2904 \g@addto@macro\do@subst@correction
2905   {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2906     \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` the patched `\pickup@font` would break the accent if different fonts are used for the base character and the accent. Fortunately,  $\LaTeX$  takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

```

2907 \let\MT@orig@add@accent\add@accent
2908 \def\add@accent#1#2{%
2909   \MT@ltx@pickupfont
2910   \MT@orig@add@accent{#1}{#2}%
2911   \MT@MT@pickupfont
2912 }%
2913 (/package)
2914 }
2915 (plain)\relax
2916 (*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.

```

2917 \def\MT@check@font{\MT@exp@one@n\MT@incl@list\MT@font\MT@font@list}

```

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

```

2918 \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).

```

2919 \def\MT@register@subst@font{\MT@exp@one@n\MT@incl@list\font@name\MT@font@list}

```

```
2920 \ifMT@inlist@else\xdef\MT@font@list{\MT@font@list\font@name,}\fi}
```

#### 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.

```
2921 \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.

```
2922 \def\MT@check@font@cx{%
2923   \MT@if@true
2924   \MT@map@clist@c\MT@active@features{%
2925     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2926     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2927     \ifMT@inlist@
2928       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2929     \else
2930       \MT@if@false
2931     \fi
2932   }%
2933   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2934 }
```

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

```
2935 \def\MT@register@subst@font@cx{%
2936   \MT@map@clist@c\MT@active@features{%
2937     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
2938     \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2939     \ifMT@inlist@ \else
2940       \MT@exp@cs\MT@xadd
2941       {MT@##1@\csname MT@##1@context\endcsname font@list}%
2942       {\font@name,}%
2943     \fi
2944   }%
2945 }
```

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

```
2946 \def\MT@register@font@cx{%
2947   \MT@map@clist@c\MT@active@features{%
2948     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2949     \MT@exp@cs\MT@xadd
2950     {MT@##1@\csname MT@##1@context\endcsname font@list}%
2951     {\MT@font,}%
2952     \def\@tempa{##1}%
2953     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2954   \fi
2955   }%
2956 }
```

`\MT@maybe@rem@from@list` Recurse through all context font lists of the document and remove the font, unless it's the current context.

```
2957 \def\MT@maybe@rem@from@list#1{%
2958   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2959     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2960     \MT@font \csname MT@\@tempa @#1font@list\endcsname
2961   }%
2962 }
```

`\microtypecontext` The user may change the context, so that different setups are possible. This is especially useful for multi-lingual documents.

Inside the preamble, it shouldn't actually do anything but remember it for later.

```
2963 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}
2964 \MT@addto@setup{%
2965   \DeclareRobustCommand\microtypecontext[1]{%
2966     \MT@setup@contexts
2967     \let\MT@reset@context\relax
```

We need to ensure that math fonts are set up anew.

```
2968   \MT@glet\glb@currsizel@empty
2969   \setkeys{MTC}{#1}%
2970   \selectfont
2971   \MT@reset@context
2972 }%
2973 }
```

`\textmicrotypecontext` This is just a wrapper around `\microtypecontext`.

```
2974 \DeclareRobustCommand\textmicrotypecontext[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@`

```
2975 \def\MT@reset@context@{%
2976   \MT@vinfo{<<< Resetting contexts\on@line
2977   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2978   <debug>           /\MT@tr@context/\MT@kn@context/\MT@sp@context
2979   }%
2980   \selectfont
2981 }
```

`\MT@setup@contexts` The first time `\microtypecontext` is called, we initialise the context lists and redefine the commands used in `\pickup@font`.

```
2982 \def\MT@setup@contexts{%
2983   \MT@map@clist@c\MT@active@features
2984   {\MT@glet@c{MT@##1@font@list}\MT@font@list}%
2985   \MT@glet\MT@check@font\MT@check@font@cx
2986   \MT@glet\MT@register@font\MT@register@font@cx
2987   \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
2988   \MT@glet\MT@setup@contexts\relax
2989 }
```

Define context keys.

```
2990 \MT@map@clist@c\MT@features@long{%
2991   \define@key{MTC}{#1}[]{}%
2992   \edef@tempb{\@nameuse{MT@rbba@#1}}%
2993   \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2994   \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).

```
2995   \MT@ifempty{#1}{\def\MT@val{0}}{\def\MT@val{#1}}%
2996   \MT@exp@cs@ifx{MT@\@tempb @context}\MT@val
2997   <debug> \MT@dinfo{1}{>>> no change of #1 context: `'\MT@val'}%
2998   \else
2999     \MT@vinfo{>>> Changing #1 context to `'\MT@val'\MessageBreak\on@line
3000   <debug>   \space(previous: `'\@nameuse{MT@\@tempb @context}')%
3001     }%
3002   \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

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

```
3003   \MT@glet@nn{MT@reset@\@tempb @codes}\MT@reset@\@tempb @codes}%
```

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

```

3004     \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
3005     \MT@val \csname MT@\@tempb @doc@contexts\endcsname
3006     \ifMT@inlist@ \else
3007     \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}}%
3008 (debug) \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
3009     \fi
3010     \MT@edef@n{MT@\@tempb @context}{\MT@val}%
3011     \fi
3012     \fi
3013 }%
3014 }

```

We also allow the activate shortcut.

```

3015 \define@key{MTC}{activate}[]{}%
3016 \setkeys{MT}{protrusion=#1}%
3017 \setkeys{MT}{expansion=#1}%
3018 }

```

\MT@pr@context     Initialise the contexts.

```

\MT@ex@context 3019 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
\MT@tr@context 3020 \MT@def@n{MT@#1@context}{@}%
\MT@sp@context 3021 \MT@def@n{MT@#1@doc@contexts}{{@}}%
3022 }
\MT@kn@context 3023 \let\MT@extra@context\@empty

```

\MT@pr@doc@contexts

\MT@ex@doc@contexts

\MT@tr@doc@contexts

\MT@sp@doc@contexts

\MT@kn@doc@contexts

\DeclareMicrotypeSet

\MT@extra@context

\DeclareMicrotypeSet\*

## 14.3 Configuration

### 14.3.1 Font sets

Calling this macro will create a comma list for every font attribute of the form: `\MT(feature)list@(attribute)@(set name)`. If the optional argument is empty, lists for all available features will be created.

The third argument must be a list of key=value pairs. If a font attribute is not specified, we define the corresponding list to `\relax`, so that it does not constitute a constraint.

```

3024 \def\DeclareMicrotypeSet{%
3025   \MT@begin@catcodes
3026   \ifstar
3027   \MT@DeclareSetAndUseIt
3028   \MT@DeclareSet
3029 }

```

\MT@DeclareSet

```

3030 \newcommand\MT@DeclareSet[3][]{%
3031   \MT@ifempty{#1}{%
3032     \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}%
3033   }%
3034   \MT@map@clist@n{#1}{%
3035     \MT@ifempty{#1}\relax%
3036     \MT@is@feature{##1}{set declaration `#2'}%
3037     \MT@exp@one@n\MT@declare@sets
3038     {\csname MT@rbba@##1\endcsname}{#2}{#3}%
3039   }%
3040   }%
3041 }%
3042 }%
3043 \MT@end@catcodes
3044 }

```

\MT@DeclareSetAndUseIt

```

3045 \newcommand\MT@DeclareSetAndUseIt[3][]{%
3046   \MT@DeclareSet[#1]{#2}{#3}%

```

```

3047 \UseMicrotypeSet[#1]{#2}%
3048 }

\MT@curr@set@name    We need to remember the name of the set currently being declared.
3049 \let\MT@curr@set@name\empty

\MT@declare@sets    Define the current set name and parse the keys.
3050 \def\MT@declare@sets#1#2#3{%
3051   \def\MT@curr@set@name{#2}%
3052   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
3053     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set ~\MT@curr@set@name'}%
3054     \MT@map@clist@n{font,encoding,family,series,shape,size}{%
3055       \MT@gl@et@nc{MT@#1@list@##1@\MT@curr@set@name}\@undefined
3056     }%
3057   }%
3058   \MT@gl@et@nc{MT@#1@set@@\MT@curr@set@name}\@empty
3059   (debug)\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set ~\MT@curr@set@name'}%
3060   \setkeys{MT@#1@set}{#3}%
3061 }

\MT@define@set@key@    <#1> = font axis, <#2> = feature.
3062 \def\MT@define@set@key@#1#2{%
3063   \define@key{MT@#2@set}{#1}[]%
3064   \MT@gl@et@nc{MT@#2@list@#1@\MT@curr@set@name}\@empty
3065   \MT@map@clist@n{##1}{%
3066     \KV@sp@def\MT@val{###1}%
3067     \MT@get@highlevel{#1}%

    We do not add the expanded value to the list ...
3068     \MT@exp@two@n@g@addto@macro
3069     {\csname MT@#2@list@#1@\MT@curr@set@name\expandafter\endcsname}%
3070     {\MT@val,}%
3071   }%

    ... but keep in mind that the list has to be expanded at the end of the preamble.
3072   \expandafter@g@addto@macro\expandafter\MT@font@sets
3073   \csname MT@#2@list@#1@\MT@curr@set@name\endcsname
3074   (debug)\MT@dinfo{n1}{1}{-- #1: \@nameuse{MT@#2@list@#1@\MT@curr@set@name}}%
3075   }%
3076 }

\MT@get@highlevel    Saying, for instance, 'family=rm*' or 'shape=bf*' will expand to \rmdefault resp.
                    \bfdefault.
3077 \def\MT@get@highlevel#1{%
3078   \expandafter\MT@test@ast\MT@val*\@nil\relax{%

    And 'family = *' will become \familydefault.
3079   \MT@ifempty@tempa{\def@tempa{#1}}\relax

    Test whether the command is actually defined.
3080   \MT@ifdefined@n@TF{\@tempa default}%
3081   {\edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}}%
3082   {\MT@warning{\@backslashchar\@tempa default' is not a defined command.\MessageBreak
3083     Ignoring `#1 = {\@tempa*}' in font set\MessageBreak~\MT@curr@set@name'}%
3084   \let\MT@val\@empty}%

    In contrast to earlier version, these values will not be expanded immediately but at
    the end of the preamble.
3085   }%
3086 }

\MT@test@ast    It the last character is an asterisk, execute the second argument, otherwise the first
                one.
3087 \def\MT@test@ast#1*#2\@nil{%

```

```

3088 \def\@tempa{#1}%
3089 \MT@ifempty{#2}%
3090 }

```

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

```

3091 \let\MT@font@sets\@empty
3092 \def\MT@fix@font@set#1{%
3093   \MT@ifdefined@c@T{#1}{%
3094     \xdef#1{#1}%
3095     \ifMT@fontspec
3096       \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
3097     \fi
3098     \global\@onelevel@sanitize#1%
3099   }%
3100 }

```

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

```

3101 \def\MT@define@set@key@size#1{%
3102   \define@key{MT@#1@set}{size}[]{}%
3103   \MT@map@clist@n{##1}{%
3104     \def\MT@val{###1}%
3105     \expandafter\MT@get@range\MT@val--\@nil
3106     \ifx\MT@val\relax \else
3107       \MT@exp@cs\MT@xadd
3108       {MT@#1list@size@MT@curr@set@name}%
3109       {{{\MT@lower}{\MT@upper}\relax}}%
3110     \fi
3111   }%
3112   <debug>\MT@dinfo@n1{1}{-- size: \@nameuse{MT@#1list@size@MT@curr@set@name}}%
3113   }%
3114 }

```

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 3115 \def\MT@get@range#1-#2-#3\@nil{%
3116   \MT@ifempty{#1}{%
3117     \MT@ifempty{#2}{%
3118       \let\MT@val\relax
3119     }%
3120     \def\MT@lower{0}%
3121     \def\MT@val{#2}%
3122     \MT@get@size
3123     \edef\MT@upper{\MT@val}%
3124   }%
3125 }%
3126 \def\MT@val{#1}%
3127 \MT@get@size
3128 \ifx\MT@val\relax \else
3129   \edef\MT@lower{\MT@val}%
3130   \MT@ifempty{#2}{%
3131     \MT@ifempty{#3}%
3132     {\def\MT@upper{-1}}%

```

2048 pt is T<sub>E</sub>X's maximum font size.

```

3133   {\def\MT@upper{2048}}%
3134   }%
3135   \def\MT@val{#2}%

```

```

3136     \MT@get@size
3137     \ifx\MT@val\relax \else
3138       \MT@ifdim\MT@lower>\MT@val{%
3139         \MT@error{%
3140           Invalid size range (\MT@lower\space > \MT@val) in font set
3141           ~\MT@curr@set@name'.\MessageBreak Swapping sizes}{}%
3142         \edef\MT@upper{\MT@lower}%
3143         \edef\MT@lower{\MT@val}%
3144       }{%
3145         \edef\MT@upper{\MT@val}%
3146       }%
3147       \MT@ifdim\MT@lower=\MT@upper
3148       {\def\MT@upper{-1}}%
3149       \relax
3150     \fi
3151   }%
3152 \fi
3153 }%
3154 }

```

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

```

3155 \def\MT@get@size{%
    A single star would mean \sizedefault, which doesn't exist, so we define it to be
    \normalsize.
3156   \if*\MT@val\relax
3157     \def\@tempa{\normalsize}%
3158   \else
3159     \MT@let@cn\@tempa{\MT@val}%
3160   \fi
3161   \ifx\@tempa\relax \else

```

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).

```

3162   \begin@group
3163     \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
3164   \@tempa\@nil
3165   \fi

```

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

```

3166   \MT@ifdimen\MT@val{%
3167     \@defaultunits\@tempdima\MT@val pt\relax\@nnil
3168     \edef\MT@val{\strip@pt\@tempdima}%
3169   }{%
3170     \MT@warning{Could not parse font size ~\MT@val'\MessageBreak
3171       in font set ~\MT@curr@set@name'}%
3172     \let\MT@val\relax
3173   }%
3174 }

```

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

```

3175 \def\MT@define@set@key@font#1{%
3176   \define@key{MT@#1@set}{font}[]{}%
3177   \MT@gl@et@nc{MT@#1@list@font\MT@curr@set@name}\@empty
3178   \MT@map@clist@n{##1}{%
3179     \def\MT@val{####1}%
3180     \MT@ifstreq\MT@val*\{def\MT@val{*/*/*/*/}\relax
3181     \expandafter\MT@get@font\MT@val////\@nil
3182     \MT@exp@two@n@g@addto@macro
3183     {\csname MT@#1@list@font\MT@curr@set@name\expandafter\endcsname}%

```

```

3184     {\MT@val,}%
3185   }%
3186   \expandafter\g@addto@macro\expandafter\MT@font@sets
3187     \csname MT@#1list@font@MT@curr@set@name\endcsname
3188   (debug)\MT@dinfol{1}{-- font: \@nameuse{MT@#1list@font@MT@curr@set@name}}%
3189   }%
3190 }

```

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

```

3191 \def\MT@get@font#1/#2/#3/#4/#5/#6\nil{%
3192   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
3193   \ifx\MT@val\relax\def\MT@val{0}\fi
3194   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
3195   \let\MT@val\@tempb
3196 }

```

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

```

3197 \def\MT@get@font@#1#2#3#4#5#6{%
3198   \let\@tempb\@empty
3199   \def\MT@temp{#1/#2/#3/#4/#5}%
3200   \MT@get@axis{encoding}{#1}%
3201   \MT@get@axis{family}{#2}%
3202   \MT@get@axis{series}{#3}%
3203   \MT@get@axis{shape}{#4}%
3204   \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
3205   \MT@ifempty{#5}{%
3206     \MT@warn@axis@empty{size}{\string\normalsize}%
3207     \def\MT@val{*}%
3208   }{%
3209     \def\MT@val{#5}%
3210   }%
3211   \MT@get@size
3212 }

```

`\MT@get@axis`

```

3213 \def\MT@get@axis#1#2{%
3214   \def\MT@val{#2}%
3215   \MT@get@highlevel{#1}%
3216   \MT@ifempty\MT@val{%
3217     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
3218     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
3219   }\relax
3220   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
3221 }

```

`\MT@warn@axis@empty`

```

3222 \def\MT@warn@axis@empty#1#2{%
3223   \MT@warning{#1 axis is empty in font specification\MessageBreak
3224     ~\MT@temp'. Using ~#2' instead}%
3225 }

```

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

```

3226 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
3227   \MT@define@set@key@{encoding}{#1}%
3228   \MT@define@set@key@{family}{#1}%
3229   \MT@define@set@key@{series}{#1}%
3230   \MT@define@set@key@{shape}{#1}%
3231   \MT@define@set@key@size{#1}%
3232   \MT@define@set@key@font{#1}%
3233 }

```

`\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.

```

3234 \def\UseMicrotypeSet{%
3235   \MT@begin@catcodes
3236   \MT@UseMicrotypeSet
3237 }

```

\MT@UseMicrotypeSet

```

3238 \newcommand*\MT@UseMicrotypeSet[2] [] {%
3239   \MT@ifempty{#1}{%
3240     \MT@map@clist@c\MT@features{\MT@use@set{##1}{#2}}%
3241   }{%
3242     \MT@map@clist@n{#1}{%
3243       \MT@ifempty{##1}\relax{%
3244         \MT@is@feature{##1}{activation of set `#2'}%
3245         \MT@exp@one@n\MT@use@set
3246         {\csname MT@rbba@##1\endcsname}{#2}%
3247       }%
3248     }%
3249   }}%
3250 }%
3251 \MT@end@catcodes
3252 }

```

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

```

\MT@ex@setname 3253 \def\MT@use@set#1#2{%
\MT@tr@setname 3254 \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
3255   \MT@xdef@n{MT@#1@setname}{#2}%
\MT@sp@setname 3256 }{%
\MT@kn@setname 3257 \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
3258   \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
\MT@use@set 3259 }%
3260 \MT@error{%
3261   The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
3262   Using set ` \@nameuse{MT@#1@setname}' instead}}%
3263 }%
3264 }

```

\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.

```

3265 \def\DeclareMicrotypeSetDefault{%
3266   \MT@begin@catcodes
3267   \MT@DeclareMicrotypeSetDefault
3268 }

```

\MT@DeclareMicrotypeSetDefault

```

3269 \newcommand*\MT@DeclareMicrotypeSetDefault[2] [] {%
3270   \MT@ifempty{#1}{%
3271     \MT@map@clist@c\MT@features{\MT@set@default@set{##1}{#2}}%
3272   }{%
3273     \MT@map@clist@n{#1}{%
3274       \MT@ifempty{##1}\relax{%
3275         \MT@is@feature{##1}{declaration of default set `#2'}%
3276         \MT@exp@one@n\MT@set@default@set
3277         {\csname MT@rbba@##1\endcsname}{#2}%
3278       }%
3279     }%
3280   }}%
3281 }%
3282 \MT@end@catcodes
3283 }

```

\MT@default@pr@set

```

\MT@default@ex@set 3284 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 3285 \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
3286 <debug>\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
\MT@default@sp@set 3287 \MT@xdef@n{MT@default@#1@set}{#2}%
\MT@default@kn@set
\MT@set@default@set

```

```

3288 }{%
3289 \MT@error{%
3290   The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
3291   Cannot make it the default set. Using set\MessageBreak `all' instead}}%
3292 \MT@xdef@n{MT@default@#1@set}{all}%
3293 }%
3294 }

```

### 14.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version `\MT@variants` appends to the list.

```

3295 \let\MT@variants\@empty
3296 \def\DeclareMicrotypeVariants{%
3297   \MT@begin@catcodes
3298   \ifstar
3299     \MT@DeclareVariants
3300   {\let\MT@variants\@empty\MT@DeclareVariants}%
3301 }

```

`\MT@DeclareVariants`

```

3302 \def\MT@DeclareVariants#1{%
3303   \MT@map@clist@n{#1}%
3304   \def\@tempa{##1}%
3305   \@onelevel@sanitize\@tempa
3306   \xdef\MT@variants{\MT@variants{\@tempa}}%
3307 }%
3308 \MT@end@catcodes
3309 }

```

`\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.

```

3310 \def\DeclareMicrotypeAlias{%
3311   \MT@begin@catcodes
3312   \MT@DeclareMicrotypeAlias
3313 }

```

`\MT@DeclareMicrotypeAlias`

```

3314 \newcommand*\MT@DeclareMicrotypeAlias[2]{%
3315   \def\@tempb{#2}%
3316   \@onelevel@sanitize\@tempb
3317   \MT@ifdefined@n{T{MT@#1@alias}}{%
3318     \MT@warning{Alias font family `@\@tempb' will override
3319       alias `@\nameuse{MT@#1@alias}'\MessageBreak
3320       for font family `#1'}}%
3321   \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.

```

3322   \MT@ifdefined@c{T\MT@family}%
3323   <debug>\MT@dinfo{1}{Activating alias font `@\@tempb' for `@\MT@family'}%
3324   \MT@gl@et\MT@familyalias\@tempb
3325   }%
3326   \MT@end@catcodes
3327 }

```

`\LoadMicrotypeFile` May be used to load a configuration file manually.

```

3328 \def\LoadMicrotypeFile#1{%
3329   \edef\@tempa{\zap@space#1 \@empty}%
3330   \@onelevel@sanitize\@tempa
3331   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list

```

```

3332 \ifMT@inlist@
3333 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
3334 \else
3335 \MT@xadd\MT@file@list{\@tempa,}%
3336 \MT@begin@catcodes
3337 \InputIfFileExists{mt-\@tempa.cfg}{%
3338 \edef\MT@curr@file{mt-\@tempa.cfg}%
3339 \MT@vinfo{... Loading configuration file \MT@curr@file}%
3340 }{%
3341 \MT@warning{Configuration file mt-\@tempa.cfg\MessageBreak
3342 does not exist}%
3343 }%
3344 \MT@end@catcodes
3345 \fi
3346 }
3347 </package>
3348 </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@nl@setname` The optional argument may be used to disable selected ligatures only.

```

\MT@nl@ligatures 3349 <*pdfTeX-def|luatex-def>
3350 <pdfTeX-def>\MT@requires@pdfTeX5{
3351 \def\DisableLigatures{%
3352 \MT@begin@catcodes
3353 \MT@DisableLigatures
3354 }
3355 \newcommand*\MT@DisableLigatures[2] [] {%
3356 \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
3357 \xdef\MT@active@features{\MT@active@features,nl}%
3358 \global\MT@no@ligaturestrue
3359 \MT@declare@sets{nl}{no ligatures}{#2}%
3360 \gdef\MT@nl@setname{no ligatures}%
3361 \MT@end@catcodes
3362 }
3363 <pdfTeX-def>}{
3364 </pdfTeX-def|luatex-def>

```

If pdfTeX is too old, we throw an error.

```

3365 <*pdfTeX-def|xetex-def>
3366 \renewcommand*\DisableLigatures[2] [] {%
3367 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
3368 with pdfTeX version 1.30 or newer.\MessageBreak
3369 Ignoring \string\DisableLigatures}{%
3370 <pdfTeX-def> Upgrade
3371 <xetex-def> Use
3372 pdfTeX.}%
3373 }
3374 <pdfTeX-def>}{
3375 </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.

```

3376 <*package>
3377 \def\DeclareMicrotypeBabelHook#1#2{%
3378 \MT@map@clist@n{#1}{%
3379 \KV@sp@def\@tempa{##1}%
3380 \MT@gdef@n{MT@babel@\@tempa}{##2}%

```

```

3381 }%
3382 }
3383 </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).

```

3384 <*pdfTeX-def|xetex-def|lualatex-def>
3385 \def\SetProtrusion{%
3386   \MT@begin@catcodes
3387   \MT@SetProtrusion
3388 }

```

`\MT@SetProtrusion` We want the catcodes to be correct even if this is called in the preamble.

```

\MT@pr@c@name 3389 \newcommand*\MT@SetProtrusion[3] [] {%

```

```

\MT@extra@context 3390 \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.

```

3391 \MT@set@named@keys{MT@pr@c}{#1}%
3392 <debug>\MT@dinfo{1}{creating protrusion list `~\MT@pr@c@name'}%
3393 \def\MT@permutelist{pr@c}%
3394 \setkeys{MT@cfig}{#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>`, ...

```

3395 \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.

```

3396 \MT@gdef@n{MT@pr@c@MT@pr@c@name}{#3}%
3397 \MT@end@catcodes
3398 }
3399 </pdfTeX-def|xetex-def|lualatex-def>

```

`\SetExpansion` `\SetExpansion` only differs in that it allows some extra options (stretch, shrink, step, auto).

```

3400 <*pdfTeX-def|lualatex-def>
3401 \def\SetExpansion{%
3402   \MT@begin@catcodes
3403   \MT@SetExpansion
3404 }

```

`\MT@SetExpansion`

```

\MT@ex@c@name 3405 \newcommand*\MT@SetExpansion[3] [] {%

```

```

\MT@extra@context 3406 \let\MT@extra@context\@empty

```

```

\MT@permutelist 3407 \MT@set@named@keys{MT@ex@c}{#1}%

```

```

3408 \MT@ifdefined@n@T{MT@ex@c@MT@ex@c@name @factor}{%
3409   \ifnum\c@name MT@ex@c@MT@ex@c@name @factor\endc@name > \@m
3410   \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@MT@ex@c@name @factor}
3411     too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the
3412     maximum of 1000}%
3413   \MT@gl@et@nc{MT@ex@c@MT@ex@c@name @factor}\@m
3414   \fi
3415 }%
3416 <debug>\MT@dinfo{1}{creating expansion list `~\MT@ex@c@name'}%

```

```

3417 \def\MT@permutelist{ex@c}%
3418 \setkeys{MT@cfg}{#2}%
3419 \MT@permute
3420 \MT@gdef@n{MT@ex@c@MT@ex@c@name}{#3}%
3421 \MT@end@catcodes
3422 }

```

## \SetTracking

```

3423 \def\SetTracking{%
3424 \MT@begin@catcodes
3425 \MT@SetTracking
3426 }

```

## \MT@SetTracking Third argument may be empty.

```

3427 \newcommand*\MT@SetTracking[3] [] {%
3428 \let\MT@extra@context\@empty
3429 \MT@set@named@keys{MT@tr@c}{#1}%
3430 debug\MT@dinfo{1}{creating tracking list `~\MT@tr@c@name'}%
3431 \def\MT@permutelist{tr@c}%
3432 \setkeys{MT@cfg}{#2}%
3433 \MT@permute
3434 \KV@sp@def\@tempa{#3}%
3435 \MT@ifempty\@tempa\relax{%
3436 \MT@ifint\@tempa
3437 {\MT@xdef@n{MT@tr@c@MT@tr@c@name}{\@tempa}}%
3438 {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
3439 tracking set `~\MT@curr@set@name'}}}%
3440 \MT@end@catcodes
3441 }
3442 pdfTeX-def|LaTeX-def

```

## \SetExtraSpacing

```

3443 pdfTeX-def
3444 \def\SetExtraSpacing{%
3445 \MT@begin@catcodes
3446 \MT@SetExtraSpacing
3447 }

```

## \MT@SetExtraSpacing

```

\MT@sp@c@name 3448 \newcommand*\MT@SetExtraSpacing[3] [] {%
\MT@extra@context 3449 \let\MT@extra@context\@empty
\MT@permutelist 3450 \MT@set@named@keys{MT@sp@c}{#1}%
3451 debug\MT@dinfo{1}{creating spacing list `~\MT@sp@c@name'}%
3452 \def\MT@permutelist{sp@c}%
3453 \setkeys{MT@cfg}{#2}%
3454 \MT@permute
3455 \MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}%
3456 \MT@end@catcodes
3457 }

```

## \SetExtraKerning

```

3458 \def\SetExtraKerning{%
3459 \MT@begin@catcodes
3460 \MT@SetExtraKerning
3461 }

```

## \MT@SetExtraKerning

```

\MT@kn@c@name 3462 \newcommand*\MT@SetExtraKerning[3] [] {%
\MT@extra@context 3463 \let\MT@extra@context\@empty
\MT@permutelist 3464 \MT@set@named@keys{MT@kn@c}{#1}%
3465 debug\MT@dinfo{1}{creating kerning list `~\MT@kn@c@name'}%
3466 \def\MT@permutelist{kn@c}%
3467 \setkeys{MT@cfg}{#2}%
3468 \MT@permute
3469 \MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}%

```

```

3470 \MT@end@catcodes
3471 }
3472 (/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`

```

3473 (*package)
3474 \def\MT@set@named@keys#1#2{%
3475   \def\x##1name=##2,##3\@nil{%
3476     \setkeys{#1}{name=##2}%
3477     \gdef\MT@options{##1##3}%
3478     \MT@rem@from@clist{name=}\MT@options
3479   }%
3480   \x#2,name=,\@nil
3481   \@expandtwoargs\setkeys{#1}\MT@options
3482 }

```

`\MT@define@code@key` Define the keys for the configuration lists (which are setting the codes, in pdfT<sub>E</sub>X speak).

```

3483 \def\MT@define@code@key#1#2{%
3484   \define@key{MT@#2}{#1}[]{%
3485     \@tempcnta=\@ne
3486     \MT@map@clist@n{##1}{%
3487       \KV@sp@def\MT@val{###1}%

```

Here, too, we allow for something like ‘bf\*’. It will be expanded immediately.

```

3488     \MT@get@highlevel{#1}%
3489     \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
3490     \advance\@tempcnta \@ne
3491   }%
3492 }%
3493 }

```

`\MT@define@code@key@family` Remove fontspec’s internal feature counter.

```

3494 \def\MT@define@code@key@family#1{%
3495   \define@key{MT@#1}{family}[]{%
3496     \@tempcnta=\@ne
3497     \MT@map@clist@n{##1}{%
3498       \KV@sp@def\MT@val{###1}%
3499       \MT@get@highlevel{family}%
3500       \ifMT@fontspec
3501         \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
3502       \fi
3503       \MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}%
3504       \advance\@tempcnta \@ne
3505     }%
3506   }%
3507 }

```

`\MT@define@code@key@size` `\MT@tempsize` must be in a `\csname`, so that it is at least `\relax`, not undefined.

```

3508 \def\MT@define@code@key@size#1{%
3509   \define@key{MT@#1}{size}[]{%
3510     \MT@map@clist@n{##1}{%
3511       \KV@sp@def\MT@val{###1}%
3512       \expandafter\MT@get@range\MT@val--\@nil
3513       \ifx\MT@val\relax \else
3514         \MT@exp@cs\MT@xadd\MT@tempsize{%
3515           {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
3516         \fi
3517       }%
3518     }%
3519 }

```

`\MT@define@code@key@font`

```

3520 \def\MT@define@code@key@font#1{%
3521   \define@key{MT@#1}{font}[]{%
3522     \MT@map@clist@n{##1}{%
3523       \KV@esp@def\MT@val{###1}%
3524       \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/}}\relax
3525       \expandafter\MT@get@font@and@size\MT@val///// \@nil
3526       \ifMT@fontspec
3527         \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
3528         \fi
3529         \MT@xdef@n{MT@\MT@permutelist @\@tempb\MT@extra@context}%
3530         {\csname MT@\MT@permutelist @name\endcsname}%
3531 (debug)\MT@dinfoln{1}{initialising: use list for font \@tempb=\MT@val
3532 (debug)           \ifx\MT@extra@context\@empty\else\MessageBreak
3533 (debug)           (context: \MT@extra@context)\fi}%
3534         \MT@exp@cs\MT@xadb
3535         {MT@\MT@permutelist @\@tempb\MT@extra@context @sizes}%
3536         {{{\MT@val}{\mone}{\MT@curr@set@name}}}%
3537     }%
3538 }%
3539 }

```

`\MT@get@font@and@size` Translate any asterisks and split off the size.

```

3540 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
3541   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
3542 }

3543 \MT@define@code@key{encoding}{cfg}
3544 \MT@define@code@key@family {cfg}
3545 \MT@define@code@key{series} {cfg}
3546 \MT@define@code@key{shape} {cfg}
3547 \MT@define@code@key@size {cfg}
3548 \MT@define@code@key@font {cfg}

```

`\MT@define@opt@key`

```

3549 \def\MT@define@opt@key#1#2{%
3550   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
3551     \MT@xdef@n{MT@#1@c@\MT@curr@set@name @#2}{##1}}%
3552 }

```

`\MT@listname@count` The options in the optional first argument.

```

3553 \newcount\MT@listname@count
3554 \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`).

```

3555   \define@key{MT@#1@c}{name}[]{%
3556     \MT@ifempty{##1}{%
3557       \MT@ifdefined@n@TF{MT@#1@c@\MT@curr@file/\the\inputlineno}{%
3558         \global\advance\MT@listname@count\@ne
3559         \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno
3560           (\number\MT@listname@count)}%
3561       }{%
3562         \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
3563       }%
3564     }{%
3565       \MT@edef@n{MT@#1@c@name}{##1}%
3566       \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
3567         \MT@warning{Redefining \@nameuse{MT@abbr#1} list ` \@nameuse{MT@#1@c@name}'}%
3568       }%
3569     }%
3570     \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
3571   }%
3572   \MT@define@opt@key{#1}{load}%
3573   \MT@define@opt@key{#1}{factor}%

```

```

3574 \MT@define@opt@key{#1}{preset}%
3575 \MT@define@opt@key{#1}{inputenc}%

```

Only one context is allowed. This might change in the future.

```

3576 \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
3577 }
3578 </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.

```

3579 <*pdfTeX-def|luatex-def>
3580 <pdfTeX-def>\MT@requires@pdfTeX7{
3581 \define@key{MT@ex@c}{context}[]{%
3582 \MT@ifempty{#1}\relax{%
3583 \MT@gllet\MT@copy@font\MT@copy@font@
3584 \def\MT@extra@context{#1}%
3585 }%
3586 }
3587 \MT@addto@setup{%
3588 \define@key{MT@ex@c}{context}[]{%
3589 \ifx\MT@copy@font\MT@copy@font@
3590 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3591 \else
3592 \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
3593 Ignoring `context' key\on@line}%
3594 {Either move the settings inside the preamble,\MessageBreak
3595 or load the package with the `copyfonts' option.}%
3596 \fi
3597 }%
3598 }

```

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.

```

3599 \define@key{MT@pr@c}{context}[]{%
3600 \MT@ifempty{#1}\relax{%
3601 \MT@gllet\MT@copy@font\MT@copy@font@
3602 \def\MT@extra@context{#1}%
3603 }%
3604 }
3605 \MT@addto@setup{%
3606 \define@key{MT@pr@c}{context}[]{%
3607 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3608 \ifx\MT@copy@font\MT@copy@font@\else
3609 \MT@warning@nl{If protrusion contexts don't work as expected,
3610 \MessageBreak load the package with the `copyfonts' option}%
3611 \fi
3612 }%
3613 }
3614 </pdfTeX-def|luatex-def>
3615 <*pdfTeX-def>
3616 {}{
3617 \define@key{MT@ex@c}{context}[]{%
3618 \MT@error{Expansion contexts only work with pdfTeX 1.40.4\MessageBreak
3619 or later. Ignoring `context' key\on@line}%
3620 {Upgrade pdfTeX.}%
3621 }
3622 </pdfTeX-def>
3623 <*pdfTeX-def|xetex-def>

```

```

3624 \define@key{MT@pr@c}{context}[]{%
3625 \MT@error{Protrusion contexts only work with pdftex
3626 (pdftex-def) 1.40.4\MessageBreak or later.
3627 (xetex-def) \MessageBreak or luatex.
3628 Ignoring `context' key\on@line}%
3629 (pdftex-def) {Upgrade pdftex.}%
3630 (xetex-def) {Use pdftex or luatex.}%
3631 }
3632 (/pdftex-def|xetex-def)
3633 (pdftex-def)

```

`\MT@warn@nodim`

```

3634 (*package)
3635 \def\MT@warn@nodim#1{%
3636 \MT@warning{'\@tempa' is not a dimension.\MessageBreak
3637 Ignoring it and setting values relative to\MessageBreak #1}%
3638 }
3639 (/package)

```

Protrusion codes may be relative to character width, or to any dimension.

```

3640 (*pdftex-def|xetex-def|luatex-def)
3641 \define@key{MT@pr@c}{unit}[character]{%
3642 \MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@empty
3643 \def\@tempa{#1}%
3644 \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.

```

3645 \MT@ifdimen\@tempa
3646 {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
3647 {\MT@warn@nodim{character widths}}%
3648 }%
3649 }
3650 (/pdftex-def|xetex-def|luatex-def)

```

Tracking may only be relative to a dimension.

```

3651 (*pdftex-def|luatex-def)
3652 \define@key{MT@tr@c}{unit}[1em]{%
3653 \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
3654 \def\@tempa{#1}%
3655 \MT@ifdimen\@tempa
3656 {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
3657 {\MT@warn@nodim{1em}}%
3658 \MT@gdefn{MT@tr@c@MT@curr@set@name @unit}{1em}}%
3659 }
3660 (/pdftex-def|luatex-def)

```

Spacing and kerning codes may additionally be relative to space dimensions.

```

3661 (*pdftex-def)
3662 \MT@map@clist@n{sp,kn}{%
3663 \define@key{MT@#1@c}{unit}[space]{%
3664 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
3665 \def\@tempa{##1}%
3666 \MT@ifstreq\@tempa{character}\relax{%
3667 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3668 \MT@ifstreq\@tempa{space}\relax{%
3669 \MT@ifdimen\@tempa
3670 {\MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3671 {\MT@warn@nodim{width of space}}%
3672 }%
3673 }%
3674 }%
3675 }
3676 (/pdftex-def)

```

The first argument to `\SetExpansion` accepts some more options.

```
3677 (*pdf $\textit{tex}$ -def| $\textit{luatex}$ -def)
3678 \MT@map@clist@n{stretch,shrink,step}{%
3679   \define@key{MT@ex@c}{#1}[]{%
3680     \MT@ifempty{##1}\relax{%
3681       \MT@ifint{##1}{%
```

A space terminates the number.

```
3682   \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%
3683   }{%
3684     \MT@warning{%
3685       Value `##1' for option `#1' is not a number.\MessageBreak
3686       Ignoring it}%
3687   }%
3688 }%
3689 }%
3690 }
3691 \define@key{MT@ex@c}{auto}[true]{%
3692   \def\@tempa{#1}%
3693   \csname if\@tempa\endcsname
```

Don't use autoexpand for pdf $\textit{T}_{\textit{E}}\textit{X}$  version older than 1.20.

```
3694 (*pdf $\textit{tex}$ -def)
3695   \MT@requires@pdf $\textit{t}_{\textit{E}}\textit{X}$ 4{%
3696     \MT@gdef@n{MT@ex@c@MT@curr@set@name @auto}{autoexpand}%
3697   }{%
3698     \MT@warning{pdf $\textit{t}_{\textit{E}}\textit{X}$  too old for automatic font expansion}%
3699   }
3700 \textit{pdf $\textit{t}_{\textit{E}}\textit{X}$ -def}
3701   \else
3702 (*pdf $\textit{tex}$ -def)
3703   \MT@requires@pdf $\textit{t}_{\textit{E}}\textit{X}$ 4{%
3704     \MT@glet@nc{MT@ex@c@MT@curr@set@name @auto}\@empty
3705   }\relax
3706 \textit{pdf $\textit{t}_{\textit{E}}\textit{X}$ -def}
3707 (* $\textit{luatex}$ -def)
3708   \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
3709      $\textit{luatex}$ }%
3710 \textit{luatex}-def}
3711   \fi
3712 }
```

Tracking: Interword spacing and outer kerning. The variant with space just in case `\SetTracking` is called inside an argument (e.g., to `\IfFileExists`).

```
3713 \MT@define@opt@key{tr}{spacing}
3714 \MT@define@opt@key{tr}{outerspacing}
3715 \MT@define@opt@key{tr}{outerkerning}
```

Which ligatures should be disabled?

```
3716 \define@key{MT@tr@c}{no ligatures}[]%
3717   {\MT@xdef@n{MT@tr@c@MT@curr@set@name @no ligatures}{#1}}
3718 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
3719 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
3720 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{no ligatures={#1}}}
3721 \textit{pdf $\textit{t}_{\textit{E}}\textit{X}$ -def}| $\textit{luatex}$ -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.

```
3722 (*package)
3723 \renewcommand*\DeclareCharacterInheritance[1] [] {%
3724   \let\MT@extra@context\@empty
3725   \let\MT@extra@inputenc\@undefined
3726   \let\MT@inh@feat\@empty
3727   \setkeys{MT@inh@}{#1}%
3728   \MT@begin@catcodes
3729   \MT@set@inh@list
3730 }
```

`\MT@set@inh@list` Safe category codes.

```
3731 \def\MT@set@inh@list#1#2{%
3732   \MT@ifempty\MT@inh@feat{%
3733     \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{##2}}}%
3734   }{%
3735     \MT@map@clist@c\MT@inh@feat{{%
3736       \KV@sp@def\@tempa{##1}%
3737       \MT@ifempty\@tempa\relax{%
3738         \MT@exp@one@n\MT@declare@char@inh
3739         {\csname MT@rbba@\@tempa@endcsname}{##1}{##2}%
3740       }%
3741     }}%
3742   }%
3743   \MT@end@catcodes
3744 }
```

The keys for the optional argument.

```
3745 \MT@map@clist@c\MT@features@long{%
3746   \define@key{MT@inh@}{#1} [] {\edef\MT@inh@feat{\MT@inh@feat#1,}}%
3747   \define@key{MT@inh@}{inputenc} {\def\MT@extra@inputenc{#1}}
```

`\MT@declare@char@inh` The lists cannot be given a name by the user.

```
3748 \def\MT@declare@char@inh#1#2#3{%
3749   \MT@edef@n{MT@#1@inh@name}%
3750   {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3751   \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3752   \MT@ifdefined@c@T\MT@extra@inputenc{%
3753     \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}%
3754   }(debug)\MT@dinfor{1}{creating inheritance list ` \@nameuse{MT@#1@inh@name}' }%
3755   \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3756   \def\MT@permutelist{#1@inh}%
3757   \setkeys{MT@inh}{#2}%
3758   \MT@permute
3759 }
```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations. We can reuse the key setup from the configuration lists (`\Set...`).

```
3760 \MT@define@code@key{encoding}{inh}
3761 \MT@define@code@key@family {inh}
3762 \MT@define@code@key@series {inh}
3763 \MT@define@code@key@shape {inh}
3764 \MT@define@code@key@size {inh}
3765 \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 ex-

ecuted only once, namely the first time this inheritance list is encountered (in `\MT@set@{feature}@codes`).

```
3766 \def\MT@inh@do#1,{%
3767   \ifx\relax#1\@empty \else
3768     \MT@inh@split #1==\relax
3769     \expandafter\MT@inh@do
3770   \fi
3771 }
```

`\MT@inh@split` Only gather the inheriting characters here. Their codes will actually be set in `\MT@set@{feature}@codes`.

```
3772 </package>
3773 <{*pdfTeX-def|xetex-def|luatex-def}>
3774 \def\MT@inh@split#1=#2=#3\relax{%
3775   \def\@tempa{#1}%
3776   \ifx\@tempa\@empty \else
3777     \MT@get@slot
3778     <pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3779     <xetex-def> \ifx\MT@char\@empty\else
3780       \let\MT@val\MT@char
3781       \MT@map@clist@n{#2}{%
3782         \def\@tempa{#1}%
3783         \ifx\@tempa\@empty \else
3784           \MT@get@slot
3785         <pdfTeX-def|luatex-def> \ifnum\MT@char > \m@ne
3786         <xetex-def> \ifx\MT@char\@empty\else
3787           \MT@exp@cs\MT@xadd\MT@inh@\MT@listname @\MT@val @{{\MT@char}}%
3788         \fi
3789       \fi
3790     }%
3791     <debug>\MT@dinfol{2}{children of #1 (\MT@val):
3792     <debug> \@nameuse{\MT@inh@\MT@listname @\MT@val @}}%
3793     \fi
3794   \fi
3795 }
3796 </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 attributes of the form `\MT@{list type}@/{encoding}/{family}/{series}/{shape}/{|*}` to be the expansion of `\MT@{list type}@name`, i.e., the name of the currently defined list. Size ranges are held in a separate macro called `\MT@{list type}@/{font axes}@sizes`, which in turn contains the respective `{list name}s` attached to the ranges.

```
3797 <{*package}>
3798 \def\MT@permute{%
3799   \let\MT@cnt@encoding\@ne
3800   \MT@permute@
```

Undefine commands for the next round.

```
3801   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3802   \MT@gl@t\MT@temp@size\@undefined
3803 }
3804 \def\MT@permute@{%
3805   \let\MT@cnt@family\@ne
3806   \MT@permute@@
3807   \MT@increment\MT@cnt@encoding
3808   \MT@ifdefined@n@T{\MT@temp@encoding\MT@cnt@encoding}%
3809   \MT@permute@
3810 }
3811 \def\MT@permute@@{%
3812   \let\MT@cnt@series\@ne
```

```

3813 \MT@permute@@@
3814 \MT@increment\MT@cnt@family
3815 \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3816 \MT@permute@@@
3817 }
3818 \def\MT@permute@@@{%
3819 \let\MT@cnt@shape\@ne
3820 \MT@permute@@@
3821 \MT@increment\MT@cnt@series
3822 \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3823 \MT@permute@@@
3824 }
3825 \def\MT@permute@@@{%
3826 \MT@permute@@@
3827 \MT@increment\MT@cnt@shape
3828 \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3829 \MT@permute@@@
3830 }

```

\MT@permute@@@ In order to save some memory, we can ignore unused encodings (inside the document).

```

3831 \def\MT@permute@@@{%
3832 \MT@permute@define(encoding)%
3833 \ifMT@document
3834 \ifx\MT@tempencoding\@empty \else
3835 \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
3836 {\expandafter\expandafter\expandafter\@gobble}%
3837 \fi
3838 \fi
3839 \MT@permute@@@
3840 }

```

\MT@permute@@@@

```

3841 \def\MT@permute@@@@{%
3842 \MT@permute@define{family}%
3843 \MT@permute@define{series}%
3844 \MT@permute@define{shape}%
3845 \edef\@tempa{\MT@tempencoding
3846 \/\MT@tempfamily
3847 \/\MT@tempseries
3848 \/\MT@tempshape
3849 \/\MT@ifdefined@c@T\MT@tempsize *}%

```

Some sanity checks: an encoding must be specified (unless nothing else is).

```

3850 \MT@ifstreq\@tempa{////}\relax{%
3851 \ifx\MT@tempencoding\@empty
3852 \MT@warning{%
3853 You have to specify an encoding for\MessageBreak
3854 \@nameuse{MT@abbr@\MT@permutelist} list
3855 ~\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
3856 Ignoring it}%
3857 \else
3858 \MT@ifdefined@c@TF\MT@tempsize{%

```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```

3859 \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3860 \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3861 }%
3862 \MT@exp@cs\MT@xaddb
3863 {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
3864 \MT@tempsize
3865 <debug>\MT@dinfo@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3866 <debug> sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
3867 <debug> @sizes\endcsname}%

```

```
3868 }{%
```

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.

```
3869 \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3870 \MT@ifstreq{\csname MT@MT@permutelist @\@tempa\MT@extra@context\endcsname}%
3871 {\csname MT@MT@permutelist @\csname MT@MT@permutelist @name\endcsname @load\endcsname}%
3872 \relax}%
3873 \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3874 ~\@nameuse{MT@MT@permutelist @name}' will\MessageBreak override
3875 list ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3876 for \MessageBreak font ~\@tempa'}%
3877 }%
3878 }%
3879 <debug>\MT@dinfol{1}{initialising: use list for font \@tempa
3880 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
3881 <debug> (context: \MT@extra@context)\fi}%
3882 }%
3883 \MT@xdef@n{MT@MT@permutelist @\@tempa\MT@extra@context}%
3884 {\csname MT@MT@permutelist @name\endcsname}%
3885 \fi
3886 }%
3887 }
```

`\MT@permute@define` Define the commands.

```
3888 \def\MT@permute@define#1{%
3889 \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3890 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3891 {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3892 {\MT@let@nc{MT@temp#1}\@empty}%
3893 }
```

`\MT@permute@reset` Reset the commands.

```
3894 \def\MT@permute@reset#1{%
3895 \@tempcnta=\@ne
3896 \MT@loop
3897 \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3898 \advance\@tempcnta\@ne
3899 \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3900 \iftrue
3901 \iffalse
3902 \MT@repeat
3903 }
```

`\MT@check@rlist` For every new range item in `\MT@tempsize`, check whether it overlaps with ranges in the existing list.

```
3904 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

`\MT@check@rlist@` Define the current new range and ...

```
3905 \def\MT@check@rlist@#1#2#3{%
3906 \def\@tempb{#1}%
3907 \def\@tempc{#2}%
3908 \MT@if@false
3909 \MT@exp@cs\MT@map@tlist@
3910 {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3911 \MT@check@range
3912 }
```

`\MT@check@range` ... recurse through the list of existing ranges.

```
3913 \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.

```
3914 \def\MT@check@range@#1#2#3{%
```

```
3915 \MT@ifdim{#2}=\m@ne{%
3916 \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
3917 \MT@ifdim\@tempb={#1}\MT@iftrue\relax
3918 }{%
```

- Item in list is a simple size, new item is a range.

```
3919 \MT@ifdim\@tempb>{#1}\relax{%
3920 \MT@ifdim\@tempc>{#1}{%
3921 \MT@iftrue
3922 \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3923 }\relax
3924 }%
3925 }%
3926 }{%
3927 \MT@ifdim\@tempc=\m@ne{%
```

- Item in list is a range, new item is a simple size.

```
3928 \MT@ifdim\@tempb<{#2}{%
3929 \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
3930 }\relax
3931 }{%
```

- Both items are ranges.

```
3932 \MT@ifdim\@tempb<{#2}{%
3933 \MT@ifdim\@tempc>{#1}{%
3934 \MT@iftrue
3935 \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3936 }\relax
3937 }\relax
3938 }%
3939 }%
3940 \ifMT@if@
3941 \MT@ifstreq{#3}%
3942 {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname @load\endcsname}%
3943 \relax{%
3944 \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
3945 ~\@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
3946 list ~#3' for font \@tempa,\MessageBreak size \@tempb}%
3947 }%
```

If we've already found a conflict with this item, we can skip the rest of the list.

```
3948 \expandafter\MT@tlist@break
3949 \fi
3950 }
```

## 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 3951 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3952 \newif\ifMT@opt@auto
3953 \newif\ifMT@opt@DVI

\MT@optwarn@admissible Some warnings.
3954 \def\MT@optwarn@admissible#1#2{%
3955 \MT@warning@n1{`#1' is not an admissible value for option\MessageBreak
```

```

3956         `#2'. Assuming `false'`%
3957 }

```

`\MT@optwarn@nan`

```

3958 /package
3959 *package|letterspace
3960 plain\MT@requires@latex1{
3961 \def\MT@optwarn@nan#1#2{%
3962 \MT@warning@n1{Value `#1' for option `#2' is not a\MessageBreak number.
3963 Using default value of \number\@nameuse{MT@#2@default}}%
3964 }
3965 plain}\relax
3966 /package|letterspace
3967 *package

```

`\MT@opt@def@set`

```

3968 \def\MT@opt@def@set#1{%
3969 \MT@ifdefined@n@TF{MT@\@tempb @set@\MT@val}{%
3970 \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
3971 }{%
3972 \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
3973 \MT@warning@n1{The #1 set `#1' is undeclared.\MessageBreak
3974 Using set `#1' instead}%
3975 }%
3976 }

```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a *<set name>*.

```

3977 \MT@map@clist@n{protrusion,expansion}{%
3978 \define@key{MT}{#1}[true]{%
3979 \csname MT@opt@#1true\endcsname
3980 \MT@map@clist@n{##1}{%
3981 \KV@sp@def\MT@val{###1}%
3982 \MT@ifempty\MT@val\relax{%
3983 \csname MT@#1true\endcsname
3984 \edef\@tempb{\csname MT@rba@#1\endcsname}%
3985 \MT@ifstreq\MT@val{true}\relax
3986 }%
3987 \MT@ifstreq\MT@val{false}{%
3988 \csname MT@#1false\endcsname
3989 }{%
3990 \MT@ifstreq\MT@val{compatibility}{%
3991 \MT@let@nc{MT@\@tempb @level}\@ne
3992 }{%
3993 \MT@ifstreq\MT@val{nocompatibility}{%
3994 \MT@let@nc{MT@\@tempb @level}\tw@
3995 }{%

```

If everything failed, it should be a set name.

```

3996 \MT@opt@def@set{#1}%
3997 }%
3998 }%
3999 }%
4000 }%
4001 }%
4002 }%
4003 }%
4004 }

```

`activate` is a shortcut for protrusion and expansion.

```

4005 \define@key{MT}{activate}[true]{%
4006 \setkeys{MT}{protrusion=#1}}%
4007 \setkeys{MT}{expansion=#1}}%
4008 }

```

spacing, kerning and tracking do not have a compatibility level.

```

4009 \MT@map@clist@n{spacing,kerning,tracking}{%
4010   \define@key{MT}{#1}[true]{%
4011     \MT@map@clist@n{##1}{%
4012       \KV@sp@def\MT@val{###1}%
4013       \MT@ifempty\MT@val\relax{%
4014         \csname MT@#1true\endcsname
4015         \MT@ifstreq\MT@val{true}\relax
4016         {%
4017           \MT@ifstreq\MT@val{false}{%
4018             \csname MT@#1false\endcsname
4019           }{%
4020             \edef\@tempb{\csname MT@rba@#1\endcsname}%
4021             \MT@opt@def@set{#1}%
4022           }%
4023         }%
4024       }%
4025     }%
4026   }%
4027 }

```

`\MT@def@bool@opt` The true/false options: draft, final (may be inherited from the class options), auto, selected, babel, DVInoutput, defersetup, copyfonts.

```

4028 \def\MT@def@bool@opt#1#2{%
4029   \define@key{MT}{#1}[true]{%
4030     \def\@tempa{##1}%
4031     \MT@ifstreq\@tempa{true}\relax{%
4032       \MT@ifstreq\@tempa{false}\relax{%
4033         \MT@optwarn@admissible{##1}{#1}%
4034         \def\@tempa{false}%
4035       }%
4036     }%
4037     #2%
4038   }%
4039 }

```

Boolean options that only set the switch.

```

4040 \MT@map@clist@n{draft,selected,babel}{%
4041   \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
4042 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotruer}

```

The DVInoutput option will change `\pdfoutput` immediately to minimise the risk of confusing other packages.

```

4043 </package>
4044 <*pdfTeX-def|LaTeX-def|xetex-def>
4045 <LaTeX-def>\MT@requires@LaTeX4{\let\pdfoutput\outputmode}\relax
4046 \MT@def@bool@opt{DVInoutput}{%
4047   \csname if\@tempa\endcsname
4048 <*pdfTeX-def|LaTeX-def>
4049   \ifnum\pdfoutput>\z@ \MT@opt@DVIntrue \fi
4050   \pdfoutput\z@
4051   \else
4052     \ifnum\pdfoutput<\@ne \MT@opt@DVIntrue \fi
4053     \pdfoutput\@ne
4054 </pdfTeX-def|LaTeX-def>
4055 <xetex-def> \MT@warning@n{Ignoring `DVInoutput' option}%
4056   \fi
4057 }
4058 </pdfTeX-def|LaTeX-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.

```

4059 (*package)
4060 \MT@def@bool@opt{deferssetup}{%
4061   \csname if@tempa@endcsname \else
4062     \AtEndOfPackage{%
4063       \MT@setup@
4064       \let\MT@setup@\empty
4065       \let\MT@addto@setup@\firstofone
4066     }%
4067   \fi
4068 }
4069 </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.

```

4070 (*pdfTeX-def|luatex-def)
4071 <pdfTeX-def>\MT@requires@pdfTeX7{
4072   \MT@def@bool@opt{copyfonts}{%
4073     \csname if@tempa@endcsname
4074       \MT@gllet\MT@copy@font\MT@copy@font@
4075     \else
4076       \MT@gllet\MT@copy@font\relax
4077     \fi
4078   }
4079 <pdfTeX-def>}{
4080 </pdfTeX-def|luatex-def>
4081 (*pdfTeX-def|xetex-def)
4082 \MT@def@bool@opt{copyfonts}{%
4083   \csname if@tempa@endcsname
4084     \MT@error
4085 <pdfTeX-def>      {The pdfTeX version you are using is too old\MessageBreak
4086 <pdfTeX-def>      to use the `copyfonts' option}{Upgrade pdfTeX.}%
4087 <xetex-def>       {The `copyfonts' option does not work with xetex}
4088 <xetex-def>       {Use pdfTeX or luatex instead.}%
4089   \fi
4090 }
4091 <pdfTeX-def>}{
4092 </pdfTeX-def|xetex-def>

```

final is the opposite to draft.

```

4093 (*package)
4094 \MT@def@bool@opt{final}{%
4095   \csname if@tempa@endcsname
4096     \MT@draftfalse
4097   \else
4098     \MT@drafttrue
4099   \fi
4100 }

```

For verbose output, we redefine \MT@vinfo.

```

4101 \define@key{MT}{verbose}[true]{%
4102   \let\MT@vinfo\MT@info@n1
4103   \def\@tempa{#1}%
4104   \MT@ifstreq\@tempa{true}\relax{%

```

Take problems seriously.

```

4105   \MT@ifstreq\@tempa{errors}{%
4106     \let\MT@warning \MT@warn@err
4107     \let\MT@warning@n1\MT@warn@err
4108   }{%
4109     \let\MT@vinfo\@gobble

```

Cast warnings to the winds.

```

4110 \MT@ifstreq\@tempa{silent}{%
4111 \let\MT@warning \MT@info
4112 \let\MT@warning@nl\MT@info@nl
4113 }{%
4114 \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
4115 }%
4116 }%
4117 }%
4118 }
4119 </package>

```

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```

4120 <*package|letterspace>
4121 <plain>\MT@requires@latexl{
4122 \MT@map@clist@n{%
4123 <package> stretch,shrink,step,%
4124 letterspace}{%
4125 \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
4126 \def\@tempa{##1 }%

```

No nonsense in \MT@factor et al.? A space terminates the number.

```

4127 \MT@ifint\@tempa
4128 {\MT@edef@n{MT@#1}{\@tempa}}%
4129 {\MT@optwarn@nan{##1}{#1}}%
4130 }%
4131 }
4132 <plain>\relax
4133 </package|letterspace>

```

factor will define the protrusion factor only.

```

4134 <*package>
4135 \define@key{MT}{factor}[\MT@factor@default]{%
4136 \def\@tempa{#1 }%
4137 \MT@ifint\@tempa
4138 {\edef\MT@pr@factor{\@tempa}}
4139 {\MT@optwarn@nan{#1}{factor}}%
4140 }

```

Unit for protrusion codes.

```

4141 \define@key{MT}{unit}[character]{%
4142 \def\@tempa{#1}%
4143 \MT@ifstreq\@tempa{character}\relax{%
4144 \MT@ifdimen\@tempa
4145 {\let\MT@pr@unit\@tempa}%
4146 {\MT@warning@nl{\@tempa' is not a dimension.\MessageBreak
4147 Ignoring it and setting values relative to\MessageBreak
4148 character widths}}%
4149 }%
4150 }

```

#### 14.4.2 Loading the definition file

\MT@endinput Abort if no capable engine found.

```

4151 \let\MT@endinput\relax
4152 \ifx\MT@engine\relax
4153 \MT@warning@nl{You don't seem to be using pdftex, luatex or xetex.\MessageBreak
4154 \MT@MT' only works with these engines.\MessageBreak
4155 I will quit now}
4156 \MT@clear@options
4157 \else

```

Otherwise load the engine-specific code (as strewn across this file).

```

4158 \input{microtype-\MT@engine tex.def}
4159 \fi
4160 \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.)

```

4161 \MT@protrusiontrue
4162 </package>
4163 <*pdftex-def|luatex-def>
4164 \ifnum\pdfoutput<\@ne \else

```

Also, we only enable expansion by default if pdfT<sub>E</sub>X can expand the fonts automatically.

```

4165 <pdftex-def> \MT@requires@pdftex4{
4166   \MT@expansiontrue
4167 <pdftex-def> \MT@autottrue
4168 <pdftex-def> }\relax
4169 \fi
4170 <luatex-def>\MT@autottrue
4171 </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 to define a no-op for the regular option processing later.

```

4172 <*package>
4173 \define@key{MT}{config}[]{\relax}
4174 \def\MT@get@config#1config=#2,#3\@nil{%
4175   \MT@ifempty{#2}%
4176   {\def\MT@config@file{\MT@MT.cfg}}%
4177   {\def\MT@config@file{#2.cfg}}%
4178 }
4179 \expandafter\expandafter\expandafter\MT@get@config
4180 \csname opt@\@currname.\@currxt\endcsname,config=,\@nil

```

Load the file.

```

4181 \IfFileExists{\MT@config@file}{%
4182   \MT@info@nl{Loading configuration file \MT@config@file}%
4183   \MT@begin@catcodes
4184   \let\MT@begin@catcodes\relax
4185   \let\MT@end@catcodes\relax
4186   \let\MT@curr@file\MT@config@file
4187   \input{\MT@config@file}%
4188   \endgroup
4189 }{\MT@warning@nl{%
4190   Could not find configuration file `~\MT@config@file'!\MessageBreak
4191   This will almost certainly cause undesired results.\MessageBreak
4192   Please fix your installation}%
4193 }

```

`\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).

```

4194 \def\MT@check@active@set#1{%
4195   \MT@ifdefined@n@TF{MT@#1@setname}{%
4196     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set `~\@nameuse{MT@#1@setname}'}%
4197   }{%

```

```

4198 \MT@ifdefined@n@TF{MT@default@#1@set}{%
4199   \MT@gl@et@n{MT@#1@setname}{MT@default@#1@set}%
4200   \MT@info@n{Using default \@nameuse{MT@abbr@#1} set \@nameuse{MT@#1@setname}'}%
4201 }{%

```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set ‘@’, and issue a warning.

```

4202   \MT@gdef@n{MT@#1@setname}{@}%
4203   \MT@warning@n{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
4204     \MessageBreak Using empty set}%
4205 }%
4206 }%
4207 }

```

#### 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}
    {\gaddto@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.

```

4208 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
4209   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
4210   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
4211 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

```

#### 14.4.5 Changing options later

`\microtypesetup`  
`\MT@define@optionX` Inside the preamble, `\microtypesetup` accepts the same options as the package (unless `defersetup=false`). In the document body, it accepts the options: `protrusion`, `expansion`, `activate`, `tracking`, `spacing` and `kerning`. Specifying font sets is not allowed.

```

4212 \def\microtypesetup{\setkeys{MT}}
4213 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
4214 \end{package}
4215 \ifx\pdfTeX-def\luatex-def\ifx\pdfTeX-def\luatex-def\ifx\pdfTeX-def\luatex-def
4216 \def\MT@define@optionX#1#2{%
4217   \define@key{MTX}{#1}[true]{%

```

```

4218 \edef\@tempb{\csname MT@rba@#1\endcsname}%
4219 \MT@map@clist@n{##1}{%
4220 \KV@sp@def\MT@val{###1}%
4221 \MT@ifempty\MT@val\relax{%
4222 \@tempcnta=\m@ne
4223 \MT@ifstreq\MT@val{true}{%

```

Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

```

4224 \MT@checksetup{#1}{%
4225 \@tempcnta=\csname MT@\@tempb @level\endcsname
4226 \MT@vinfo{Enabling #1
4227 (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
4228 }%
4229 }{%
4230 \MT@ifstreq\MT@val{false}{%
4231 \@tempcnta=\z@
4232 \MT@vinfo{Disabling #1\on@line}%
4233 }{%
4234 \MT@ifstreq\MT@val{compatibility}{%
4235 \MT@checksetup{#1}{%
4236 \@tempcnta=\@ne
4237 \MT@let@nc{MT@\@tempb @level}\@ne
4238 \MT@vinfo{Setting #1 to level 1\on@line}%
4239 }%
4240 }{%
4241 \MT@ifstreq\MT@val{nocompatibility}{%
4242 \MT@checksetup{#1}{%
4243 \@tempcnta=\tw@
4244 \MT@let@nc{MT@\@tempb @level}\tw@
4245 \MT@vinfo{Setting #1 to level 2\on@line}%
4246 }%
4247 }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
4248 {Use any of `true', `false', `compatibility' or
4249 `nocompatibility'.}%
4250 }%
4251 }%
4252 }%
4253 }%
4254 \ifnum\@tempcnta>\m@ne
4255 #2\@tempcnta\relax
4256 \fi
4257 }%
4258 }%
4259 }%
4260 }

```

`\MT@checksetup` Test whether the feature wasn't disabled in the package options.

```

4261 \def\MT@checksetup#1{%
4262 \csname ifMT@#1\endcsname
4263 \expandafter\@firstofone
4264 \else
4265 \MT@error{You cannot enable #1 if it was disabled\MessageBreak
4266 in the package options}{Load microtype with #1 enabled.}%
4267 \expandafter\@gobble
4268 \fi
4269 }

4270 \MT@define@optionX{protrusion}\MT@protrudechars
4271 </pdfTeX-def|luatex-def|xetex-def>
4272 <*pdfTeX-def|luatex-def>
4273 \MT@define@optionX{expansion}\MT@adjustspacing

```

`\MT@protrudechars`

`\MT@adjustspacing`

```

4274 <*/luatex-def>
4275 \MT@requires@luatex4{
4276   \let\pdfprotrudechars\protrudechars
4277   \let\pdfadjustspacing\adjustspacing
4278 } \relax
4279 </luatex-def>
4280 \let\MT@protrudechars\pdfprotrudechars
4281 \let\MT@adjustspacing\pdfadjustspacing
4282 </pdftex-def|luatex-def>
4283 <*/xetex-def>
4284 \let\MT@protrudechars\XeTeXprotrudechars
4285 \define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
4286 </xetex-def>

```

\MT@define@optionX@ The same for tracking, spacing and kerning, which do not have a compatibility level.

```

4287 <*/pdftex-def|luatex-def>
4288 <pdftex-def>\MT@requires@pdftex6{
4289 <luatex-def>\MT@requires@luatex3{
4290   \def\MT@define@optionX@#1#2{%
4291     \define@key{MTX}{#1}[true]{%
4292       \MT@map@clist@n{##1}{%
4293         \KV@sp@def\MT@val{###1}%
4294         \MT@ifempty\MT@val\relax{%
4295           \@tempcnta=\m@ne
4296           \MT@ifstreq\MT@val{true}{%
4297             \MT@checksetup{#1}{%
4298               \@tempcnta=\@ne
4299               \MT@vinfo{Enabling #1\on@line}%
4300             }%
4301           }%
4302           \MT@ifstreq\MT@val{false}{%
4303             \@tempcnta=\z@
4304             \MT@vinfo{Disabling #1\on@line}%
4305           }{\MT@error{Value `~\MT@val' for key `~#1' not recognised}
4306             {Use either `true' or `false'}}%
4307         }%
4308       }%
4309       \ifnum\@tempcnta>\m@ne
4310         #2\relax
4311       \fi
4312     }%
4313   }%
4314 }%
4315 }

```

We cannot simply let \MT@tracking relax, since this may select the already letter-spaced font instance.

```

4316 \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
4317   \else \let\MT@tracking\MT@tracking@ \fi}
4318 <pdftex-def> \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
4319 <pdftex-def> \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
4320 <pdftex-def> \pdfappendkern\@tempcnta}
4321 }{
4322 </pdftex-def|luatex-def>
4323 <*/pdftex-def|luatex-def|xetex-def>

```

Disable for older pdfTeX versions and for XeTeX and LuaTeX.

```

4324 \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
4325 <luatex-def>
4326 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
4327 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
4328 <pdftex-def>
4329 \define@key{MTX}{activate}[true]{%
4330   \setkeys{MTX}{protrusion=#1}}%

```

```

4331 <pdfTeX-def|luatex-def> \setkeys{MTX}{expansion={#1}}%
4332 }
4333 </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.

```

4334 <*package>
4335 \let\MT@saved@setupfont\MT@setupfont

4336 \define@key{MTX}{disable}[]{%
4337   \MT@info{Inactivate `~\MT@MT' package}%
4338   \let\MT@setupfont\relax
4339 }
4340 \define@key{MTX}{enable}[]{%
4341   \MT@info{Reactivate `~\MT@MT' package}%
4342   \let\MT@setupfont\MT@saved@setupfont
4343 }
4344 </package>

```

#### 14.4.6 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

4345 <*package|letterspace>
4346 <plain>\MT@requires@latex1{
4347 \def\MT@ProcessOptionsWithKV#1{%
4348   \let@temp\relax
4349   \let\MT@temp\empty
4350 <plain> \MT@requires@latex2{
4351   \MT@map@list@c@classoptionslist{%
4352     \def\CurrentOption{##1}%
4353     \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
4354       \def\MT@temp{\MT@temp,\CurrentOption,%
4355         \@expandtwoargs\removeelement\CurrentOption
4356         \@unusedoptionlist\unusedoptionlist
4357       }%
4358     }%
4359     \edef\MT@temp{\noexpand\setkeys{#1}%
4360       {\MT@temp\@optionlist{\@currname.\@currentt}}}%

```

`plain` can handle package options.

```

4361 <*plain>
4362 }{\edef\MT@temp{\noexpand\setkeys{#1}%
4363   {\csname usepkg@options@usepkg@pkg\endcsname}}
4364 </plain>
4365 \MT@temp
4366 \MT@clear@options
4367 }

```

`\MT@getkey` For `key=val` in class options.

```

4368 \def\MT@getkey#1=#2\@nil{#1}
4369 \MT@ProcessOptionsWithKV{MT}
4370 <plain>\relax
4371 </package|letterspace>
4372 <*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).

```

4373 \MT@addto@setup{%
4374 \ifMT@draft

```

We disable most of what we've just defined in the 4374 lines above if we are running in draft mode.

```

4375 \MT@warning@nl{'draft' option active.\MessageBreak
4376         Disabling all micro-typographic extensions.\MessageBreak
4377         This might lead to different line and page breaks}%
4378 \let\MT@setupfont\relax
4379 \renewcommand*\LoadMicrotypeFile[1]{}%
4380 \renewcommand*\microtypesetup[1]{}%
4381 \renewcommand*\microtypecontext[1]{}%
4382 \renewcommand*\lssstyle{}%
4383 \else
4384 \MT@setup@PDF
4385 \MT@setup@copies

```

Fix the font sets.

```

4386 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
4387 \MT@setup@protrusion
4388 \MT@setup@expansion
4389 \MT@setup@tracking
4390 \MT@setup@wartracking
4391 \MT@setup@spacing
4392 \MT@setup@kerning
4393 \MT@setup@noligatures
4394 }
4395 </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!

```

4396 <*pdfTeX-def|luatex-def>
4397 \def\MT@setup@PDF{%
4398 \MT@info@nl{Generating \ifnum\pdfoutput<\one DVI \else PDF \fi output%
4399 \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
4400 }

```

`\MT@setup@copies` Working on font copies?

```

4401 \def\MT@setup@copies{%
4402 \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
4403 }
4404 </pdfTeX-def|luatex-def>
4405 <*xetex-def>
4406 \let\MT@setup@PDF\relax
4407 \let\MT@setup@copies\relax
4408 </xetex-def>

```

`\MT@setup@protrusion` Protrusion.

```

4409 <*pdfTeX-def|xetex-def|luatex-def>
4410 \def\MT@setup@protrusion{%
4411 \ifMT@protrusion
4412 \edef\MT@active@features{\MT@active@features,pr}%
4413 \MT@protrudechars\MT@pr@level
4414 \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
4415 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
4416 factor: \number\MT@pr@factor\fi
4417 \ifx\MT@pr@unit@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
4418 \MT@check@active@set{pr}%
4419 \else

```

```

4420 \let\MT@protrusion\relax
4421 \MT@info@n1{No character protrusion}%
4422 \fi
4423 }
4424 (/pdfTEX-def|xetex-def|luatex-def)

```

\MT@setup@expansion For DVI output, the user must have explicitly passed the expansion option to the package.

```

4425 (*pdfTEX-def|luatex-def)
4426 \def\MT@setup@expansion{%
4427 \ifnum\pdfoutput<\@ne
4428 \ifMT@opt@expansion \else
4429 \MT@expansionfalse
4430 \fi
4431 \fi
4432 \ifMT@expansion

```

Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.

```

4433 \ifnum\MT@stretch=\m@ne
4434 \let\MT@stretch\MT@stretch@default
4435 \fi

```

If shrink has not been specified, it will inherit the value from stretch.

```

4436 \ifnum\MT@shrink=\m@ne
4437 \let\MT@shrink\MT@stretch
4438 \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.

```

4439 \ifnum\MT@step=\m@ne
4440 (/pdfTEX-def) \MT@requires@pdfTEX6{%
4441 \def\MT@step{1 }%
4442 (*pdfTEX-def)
4443 }{%
4444 \ifnum\MT@stretch>\MT@shrink
4445 \ifnum\MT@shrink=\z@
4446 \@tempcnta=\MT@stretch
4447 \else
4448 \@tempcnta=\MT@shrink
4449 \fi
4450 \else
4451 \ifnum\MT@stretch=\z@
4452 \@tempcnta=\MT@shrink
4453 \else
4454 \@tempcnta=\MT@stretch
4455 \fi
4456 \fi
4457 \divide\@tempcnta 5\relax
4458 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
4459 \edef\MT@step{\number\@tempcnta\space}%
4460 }%
4461 (/pdfTEX-def)
4462 \fi
4463 \ifnum\MT@step=\z@
4464 \MT@warning@n1{The expansion step cannot be set to zero.\MessageBreak
4465 Setting it to one}%
4466 \def\MT@step{1 }%
4467 \fi

```

\MT@auto Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the

*hiz* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdf $\TeX$ ). With Lua $\TeX$ , we just leave it empty, as there’s actually no difference – non-automatic font expansion doesn’t work anymore. In Lua $\TeX$  1.0.6, the ‘autoexpand’ option seems to have been removed altogether and would trigger an error.

```
4468 \let\MT@auto@\empty
4469 \ifMT@auto
4470
```

We turn off automatic expansion if output mode is DVI and we’re running pdf $\TeX$ .

```
4471 (*pdf $\TeX$ -def)
4472 \MT@requires@pdf $\TeX$ 4{%
4473 \ifnum\pdfoutput<\@ne
4474 \ifMT@opt@auto
4475 \MT@error{%
4476 Automatic font expansion only works for PDF output.\MessageBreak
4477 However, you are creating a DVI file}
4478 {If you have created expanded fonts instances, remove `auto' from%
4479 \MessageBreak the package options. Otherwise, you have to switch
4480 off expansion\MessageBreak completely.}%
4481 \fi
4482 \MT@autofalse
4483 \else
4484 \def\MT@auto{autoexpand}%
4485 \fi
```

Also, if pdf $\TeX$  is too old.

```
4486 }{%
4487 \MT@error{%
4488 The pdf $\TeX$  version you are using is too old for\MessageBreak
4489 automatic font expansion}%
4490 {If you have created expanded fonts instances, remove `auto' from\MessageBreak
4491 the package options. Otherwise, you have to switch off expansion\MessageBreak
4492 completely, or upgrade pdf $\TeX$  to version 1.20 or newer.}%
4493 \MT@autofalse
4494 \def\MT@auto{1000 }%
4495 }%
4496 \ifMT@opt@auto
4497 \else
4498 (*pdf $\TeX$ -def)
```

No automatic expansion.

```
4499 \MT@requires@pdf $\TeX$ 4\relax{%
4500 \def\MT@auto{1000 }%
4501 }%
4502 \ifMT@opt@auto
4503 (*luat $\TeX$ -def)
4504 \ifMT@opt@auto
4505 \MT@error{Non-automatic font expansion does not work with\MessageBreak
4506 luat $\TeX$ }{Remove `auto=false' from the package options, or use pdf $\TeX$ .}%
4507 \fi
4508 \ifMT@opt@auto
4509 \fi
```

Choose the appropriate macro for selected expansion.

```
4510 \ifMT@selected
4511 \let\MT@set@ex@codes\MT@set@ex@codes@s
4512 \else
4513 \let\MT@set@ex@codes\MT@set@ex@codes@n
4514 \fi
```

Filter out stretch=0, shrink=0, since it would result in a pdf $\TeX$  error.

```
4515 \ifnum\MT@stretch=\z@
4516 \ifnum\MT@shrink=\z@
```

```

4517     \MT@warning@nl{%
4518         Both the stretch and shrink limit are set to zero.\MessageBreak
4519         Disabling font expansion}%
4520     \MT@expansionfalse
4521     \fi
4522     \fi
4523     \fi
4524     \ifMT@expansion
4525     \edef\MT@active@features{\MT@active@features,ex}%
4526     \MT@adjustspacing\MT@ex@level
4527     \MT@info@nl{\ifMT@auto A\else Non-automatic font expansion enabled
4528         (level \number\MT@ex@level),\MessageBreak
4529         stretch: \number\MT@stretch, shrink: \number\MT@shrink,
4530         step: \number\MT@step, \ifMT@selected\else non-\fi selected}%

```

`\MT@check@step`      Check whether stretch and shrink are multiples of step.

```

4531     \def\MT@check@step##1{%
4532         \@tempcnta=\csname MT@##1\endcsname
4533         \divide\@tempcnta \MT@step
4534         \multiply\@tempcnta \MT@step
4535         \ifnum\@tempcnta=\csname MT@##1\endcsname\else
4536             \MT@warning@nl{The ##1 amount is not a multiple of step.\MessageBreak
4537                 The effective maximum ##1 is \the\@tempcnta\space
4538                 (step \number\MT@step)}%
4539         \fi
4540     }%
4541     \MT@check@step{stretch}%
4542     \MT@check@step{shrink}%
4543     \MT@check@active@set{ex}%

```

Inside `\showhyphens`, font expansion should be disabled. (Since 2017/01/10, the  $\LaTeX$  format contains a different version for  $X_{\text{}}\TeX$ , but since expansion doesn't work with  $X_{\text{}}\TeX$ , we don't have to bother.)

```

4544     \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
4545         \color@begingroup\everypar{\parfillskip\z@skip
4546         \hsizemaxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4547         \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%

```

`\showhyphens`      I wonder why it's defined globally (in `lftfssbas.dtx`)?

```

4548     \gdef\showhyphens##1{\setbox0\vbox{%
4549         \color@begingroup\pdfadjustspacing\z@\everypar{\parfillskip\z@skip
4550         \hsizemaxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
4551         \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%
4552     \else
4553         \let\MT@expansion\relax
4554         \MT@info@nl{No font expansion}%
4555     \fi
4556 }
4557 </pdfTeX-def|luatex-def>
4558 < *xetex-def >
4559 \def\MT@setup@expansion{%
4560     \ifMT@expansion
4561         \ifMT@opt@expansion
4562             \MT@error{Font expansion does not work with xetex}
4563             {Use pdfTeX or luatex instead.}%
4564         \fi
4565     \fi
4566 }
4567 </xetex-def >

```

`\MT@setup@tracking`      Tracking, spacing and kerning.

```

4568 < *pdfTeX-def|luatex-def >
4569 < pdfTeX-def >\MT@requires@pdfTeX6{%
4570 < luatex-def >\MT@requires@luatex3{%

```

```

4571 \def\MT@setup@tracking{%
4572 \ifMT@tracking
4573 \edef\MT@active@features{\MT@active@features,tr}%
4574 \MT@info@nl{Tracking enabled}%
4575 \MT@check@active@set{tr}%

```

Enable protrusion for compensation at the line edges.

```

4576 \ifMT@protrusion\else\MT@protrudechars\@ne\fi
4577 \else
4578 \let\MT@tracking\relax
4579 \MT@info@nl{No adjustment of tracking}%
4580 \fi
4581 }
4582 (/pdfTeX-def|LaTeX-def)

```

`\MT@setup@spacing`

```

4583 (*pdfTeX-def)
4584 \def\MT@setup@spacing{%
4585 \ifMT@spacing
4586 \edef\MT@active@features{\MT@active@features,sp}%
4587 \pdfadjustinterwordglue\@ne
4588 \MT@info@nl{Adjustment of interword spacing enabled}%

```

The ragged2e package sets interword spaces to a fixed value without glue. mi crotypes' modifications can therefore have undesired effects. Therefore, we issue a warning.

```

4589 \MT@with@package@T{ragged2e}{%
4590 \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
4591 Adjustment of interword spacing may lead to\MessageBreak
4592 undesired results when used with `ragged2e'.\MessageBreak
4593 In this case, disable the `spacing' option}%
4594 }%
4595 \MT@check@active@set{sp}%
4596 \else
4597 \let\MT@spacing\relax
4598 \MT@info@nl{No adjustment of interword spacing}%
4599 \fi
4600 }

```

`\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`.<sup>15</sup>

```

4601 \def\MT@setup@spacing@check{%
4602 \ifMT@spacing
4603 \ifMT@babel \else
4604 \ifnum\sfcode`. > 1500
4605 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
4606 \MT@warning@nl{%
4607 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
4608 interword spacing will disable it. You might want\MessageBreak
4609 to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
4610 to your preamble}%
4611 }%
4612 \fi
4613 \fi
4614 \fi
4615 }

```

`\MT@setup@kerning`

```

4616 \def\MT@setup@kerning{%
4617 \ifMT@kerning
4618 \edef\MT@active@features{\MT@active@features,kn}%

```

15 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

```

4619     \pdfprependkern\@ne
4620     \pdfappendkern\@ne
4621     \MT@info@n1{Adjustment of character kerning enabled}%
4622     \MT@check@active@set{kn}%
4623     \else
4624     \let\MT@kerning\relax
4625     \MT@info@n1{No adjustment of character kerning}%
4626     \fi
4627   }
4628 \end{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.

```

4629 \end{pdfTEX-def}\end{luatex-def}}{
4630 \begin{luatex-def}
4631   \def\MT@setup@tracking{%
4632     \ifMT@tracking
4633       \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
4634         or newer. Switching it off}{Upgrade luatex.}%
4635       \MT@trackingfalse
4636       \MT@let@nc{MT@tracking}\relax
4637     \else
4638       \MT@info@n1{No adjustment of tracking (luatex too old)}%
4639     \fi
4640   }
4641 }
4642 \end{luatex-def}
4643 \end{pdfTEX-def}\end{xetex-def}\end{luatex-def}
4644 \def\MT@error@doesnt@work#1{%
4645   \csname ifMT@#1\endcsname
4646   \MT@error{The #1 feature only works with pdftex 1.40\MessageBreak
4647     or newer. Switching it off}
4648   \end{pdfTEX-def} \end{luatex-def} {Upgrade pdftex.}%
4649   \end{luatex-def}\end{xetex-def} {Use pdftex instead.}%
4650   \csname MT@#1false\endcsname
4651   \MT@let@nc{MT@#1}\relax
4652   \else
4653     \MT@info@n1{No adjustment of #1%
4654   \end{pdfTEX-def} \space(pdftex too old)%
4655   }%
4656   \fi
4657 }
4658 \end{pdfTEX-def}\end{xetex-def} \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
4659 \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
4660 \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
4661 \end{pdfTEX-def}
4662 \end{pdfTEX-def}\end{xetex-def}\end{luatex-def}

```

`\MT@setup@wartracking`

```

4663 \end{letterspace}\MT@addto@setup
4664 \end{pdfTEX-def}\end{luatex-def}\def\MT@setup@wartracking

```

`\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.

```

4665 \end{pdfTEX-def}\end{luatex-def}\end{letterspace}
4666 }%
4667 \end{pdfTEX-def}\end{letterspace}
4668 \ifnum\pdfoutput<\@ne
4669   \def\MT@warn@tracking@DVI{%
4670 \end{letterspace} \MT@pdf@or@lua{%
4671   \MT@warning@n1{%
4672     You are using tracking/letterspacing in DVI mode.\MessageBreak
4673     This will probably not work, unless the post-\MessageBreak

```

```

4674         processing program (dvips, dvi2pdf(x), ...) is\MessageBreak
4675         able to create the virtual fonts on the fly}%
4676 <letterspace> } \relax
4677     \MT@gllet\MT@warn@tracking@DVI\relax
4678 }%
4679 \else
4680 </pdfTeX-def|letterspace>
4681     \def\MT@warn@tracking@DVI{%
4682         \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
4683         \MT@gllet\MT@warn@tracking@DVI\relax
4684     }%
4685 <pdfTeX-def|letterspace> \fi
4686 \ifnum\MT@letterspace=\m@ne
4687     \let\MT@letterspace\MT@letterspace@default
4688 \else
4689     \MT@ls@too@large\MT@letterspace
4690 \fi
4691 }
4692 </pdfTeX-def|luatex-def|letterspace>
4693 <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.

```

4694 <*pdfTeX-def|luatex-def>
4695 \def\MT@setup@noligatures{%
4696 <pdfTeX-def> \MT@requires@pdfTeX5{%
4697     \if\MT@noligatures \else
4698         \let\MT@noligatures\relax
4699     \fi
4700 <pdfTeX-def> } \relax
4701 }
4702 </pdfTeX-def|luatex-def>
4703 <xetex-def>\let\MT@setup@noligatures\relax

```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```

4704 <*package>
4705 \MT@addto@setup{%
4706     \ifx\MT@active@features\empty \else
4707         \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
4708     \fi
4709     \MT@documenttrue
4710 }

```

`\MT@set@babel@context`     Interaction with babel.

```

4711 \def\MT@set@babel@context#1{%
4712     \MT@ifdefined@n@TF{MT@babel@#1}{%
4713         \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
4714         \expandafter\MT@exp@one@n\expandafter\microtypecontext
4715         \csname MT@babel@#1\endcsname
4716     }{%
4717         \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
4718     }%
4719 }

```

`\MT@shorthandoff`     Active characters can only be switched off if babel isn't loaded after microtype.

```

4720 \@ifpackageloaded{babel}{
4721     \def\MT@shorthandoff#1#2{%
4722         \MT@info@n1{Switching off #1 babel's active characters (#2)}%
4723         \shorthandoff{#2}}
4724 }{
4725     \def\MT@shorthandoff#1#2{%
4726         \MT@error{You must load `babel' before `'\MT@MT'}
4727         {Otherwise, `'\MT@MT' cannot switch off #1 babel's\MessageBreak

```

```

4728         active characters.}}
4729 }

```

We patch the language switching commands to enable language-dependent setup.

```

4730 \MT@addto@setup{%
4731   \ifMT@babel
4732     \ifpackage@loaded{babel}{%
4733       \MT@info@nl{Redefining babel's language switching commands}%
4734       \let\MT@orig@select@language\select@language
4735       \def\select@language#1{%
4736         \MT@orig@select@language{#1}%
4737         \MT@set@babel@context{#1}%
4738       }%
4739       \let\MT@orig@foreign@language\foreign@language
4740       \def\foreign@language#1{%
4741         \MT@orig@foreign@language{#1}%
4742         \MT@set@babel@context{#1}%
4743       }%
4744     \ifMT@kerning

```

Disable French babel's active characters.

```

4745     \MT@if@false
4746     \MT@with@babel@and@T{french} \MT@if@true
4747     \MT@with@babel@and@T{frenchb} \MT@if@true
4748     \MT@with@babel@and@T{français}\MT@if@true
4749     \MT@with@babel@and@T{canadien}\MT@if@true
4750     \MT@with@babel@and@T{acadian} \MT@if@true
4751     \ifMT@if@MT@shorthandoff{French}{:;!}\fi

```

Disable Turkish babel's active characters.

```

4752     \MT@if@false
4753     \MT@with@babel@and@T{turkish} \MT@if@true
4754     \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
4755     \fi

```

In case babel was loaded before microtype:

```

4756     \MT@set@babel@context\language@name
4757   }{%
4758     \MT@warning@nl{You did not load the babel package.\MessageBreak
4759     The `babel' option won't have any effect}%
4760   }%
4761   \fi
4762 }

```

Now we close the \fi from \ifMT@draft.

```

4763 \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.

```

4764   \selectfont}

```

\MT@curr@file This is the current file (hopefully with the correct extension).

```

4765 \edef\MT@curr@file{\jobname.tex}
4766 </package>

```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```

4767 <*package|letterspace>
4768 <plain>\MT@requires@latex1{
4769 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@ \@empty}
4770 <plain>}\relax
4771 </package|letterspace>

```

Must come at the very, very end.

```

4772 <package>\MT@if@defined@c@T\MT@setup@spacing@check

```

```
4773 <package> {\AtBeginDocument{\MT@setup@spacing@check}}
```

Restore catcodes.

```
4774 <package|letterspace>\MT@restore@catcodes
```

That was that.

## 15 Configuration files

Let's now write the font configuration files.

```
4775 (*config)
4776
```

### 15.1 Font sets

We first declare some sets in the main configuration file.

```
4777 (*m-t)
4778 %%% -----
4779 %%% FONT SETS
4780
4781 \DeclareMicrotypeSet{all}
4782 { }
4783
4784 \DeclareMicrotypeSet{allmath}
4785 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }
4786
4787 \DeclareMicrotypeSet{alltext}
4788 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU} }
4789
4790 \DeclareMicrotypeSet{allmath-nott}
4791 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
4792   family = {rm*,sf*}
4793 }
4794
4795 \DeclareMicrotypeSet{alltext-nott}
4796 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4797   family = {rm*,sf*}
4798 }
4799
4800 \DeclareMicrotypeSet{basicmath}
4801 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
4802   family = {rm*,sf*},
4803   series = {md*},
4804   size = {normalsize,footnotesize,small,large}
4805 }
4806
4807 \DeclareMicrotypeSet{basictext}
4808 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
4809   family = {rm*,sf*},
4810   series = {md*},
4811   size = {normalsize,footnotesize,small,large}
4812 }
4813
4814 \DeclareMicrotypeSet{smallcaps}
4815 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4816   shape = {sc*,si,scit}
4817 }
4818
4819 \DeclareMicrotypeSet{footnotesize}
4820 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
4821   size = {-small}
4822 }
4823
4824 \DeclareMicrotypeSet{scriptsize}
4825 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
```

```

4826     size      = {-footnotesize}
4827   }
4828
4829 \DeclareMicrotypeSet{normal font}
4830   { font = */*/*/*/* }
4831

```

The default sets.

```

4832 %%% -----
4833 %%% DEFAULT SETS
4834
4835 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4836 \DeclareMicrotypeSetDefault[expansion]{basictext}
4837 \DeclareMicrotypeSetDefault[spacing]{basictext}
4838 \DeclareMicrotypeSetDefault[kerning]{alltext}
4839 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4840

```

## 15.2 Font variants and aliases

```

4841 %%% -----
4842 %%% FONT VARIANTS AND ALIASES

```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```

4843
4844 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}

```

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 XeTeX and LuaTeX in the L<sup>A</sup>T<sub>E</sub>X format is TU, even if fontspec is not loaded.

```

4845
4846 \MT@if@false
4847 \ifx\UnicodeEncodingName\undefined\else
4848   \MT@if@streq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
4849 \fi
4850 \ifMT@fontspec\MT@if@true\fi
4851 \ifMT@if@
4852 \DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
4853   \else
4854 \DeclareMicrotypeAlias{lmr}{cmr}           % lmodern
4855 \fi

```

The Latin Modern fonts, the virtual fonts from the `ae` and `zefonts`, and the `eco` and `hfoldsty` packages (oldstyle numerals) 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.

```

4856 \DeclareMicrotypeAlias{lmsy}{cmsy}
4857 \DeclareMicrotypeAlias{lmm}{cmm}
4858 \DeclareMicrotypeAlias{aer}{cmr}           % ae
4859 \DeclareMicrotypeAlias{zer}{cmr}           % zefonts
4860 \DeclareMicrotypeAlias{cmor}{cmr}          % eco
4861 \DeclareMicrotypeAlias{hfor}{cmr}          % hfoldsty

```

The packages `pxfonts` and `txfonts` inherit Palatino and Times settings respectively, also the T<sub>E</sub>X Gyre fonts Pagella and Termes (formerly: `qfonts`).

```

4862 \DeclareMicrotypeAlias{pxr} {ppl}          % pxfonts
4863 \DeclareMicrotypeAlias{qpl} {ppl}          % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

The 'FPL Neu' fonts, a 're-implementation' of Palatino.
4864 \DeclareMicrotypeAlias{fp9x}{pplx}        % FPL Neu
4865 \DeclareMicrotypeAlias{fp9j}{pplj}        % "

The newpx package, a replacement for pxfonts.
4866 \DeclareMicrotypeAlias{zpllf}{ppl}        % newpxtext
4867 \DeclareMicrotypeAlias{zplosf}{ppl}      % "
4868 \DeclareMicrotypeAlias{zpltlf}{ppl}      % "
4869 \DeclareMicrotypeAlias{zpltosf}{ppl}     % "
4870 \DeclareMicrotypeAlias{txr} {ptm}        % txfonts

The newtx package, a replacement for txfonts.
4871 \DeclareMicrotypeAlias{ntxlf}{ptm}       % newtxtext
4872 \DeclareMicrotypeAlias{ntxosf}{ptm}     % "
4873 \DeclareMicrotypeAlias{ntxtlf}{ptm}     % "
4874 \DeclareMicrotypeAlias{ntxtosf}{ptm}    % "

The tempora package.
4875 \DeclareMicrotypeAlias{Tempora-TLF}{ptm} % tempora
4876 \DeclareMicrotypeAlias{Tempora-TOf}{ptm}% "
4877 \DeclareMicrotypeAlias{qtm} {ptm}       % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

The OpenType versions:
4878 \DeclareMicrotypeAlias{TeX Gyre Pagella}{Palatino Linotype}
4879 \DeclareMicrotypeAlias{Palatino LT Std} {Palatino Linotype}
4880 \DeclareMicrotypeAlias{Palatino}        {Palatino Linotype}
4881 \DeclareMicrotypeAlias{Asana Math}      {Palatino Linotype}

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (Times-
NewRomanMT, TimesNRSevenMT), mtm (TimesSmallTextMT); pte (TimesEuropa);
ptt (TimesTen); TimesEighteen; TimesModernEF.

The eulervm package virtually extends the Euler fonts.
4882 \DeclareMicrotypeAlias{zeur}{eur}       % Euler VM
4883 \DeclareMicrotypeAlias{zeus}{eus}      % "

MicroPress's Charter version (chmath).
4884 \DeclareMicrotypeAlias{chr} {bch}      % CH Math

The XCharter package extends the Charter fonts.
4885 \DeclareMicrotypeAlias{XCharter-TLF} {bch} % XCharter
4886 \DeclareMicrotypeAlias{XCharter-TOf}{bch} % "

The mathdesign package provides math fonts matching Bitstream Charter and URW
Garamond.
4887 \DeclareMicrotypeAlias{mdbch}{bch}     % mathdesign/Charter
4888 \DeclareMicrotypeAlias{mdugm}{ugm}    % mathdesign/URW Garamond

The garamondx package, an extension of URW Garamond, providing small caps and
oldstyle figures.
4889 \DeclareMicrotypeAlias{zgmX}{ugm}     % garamondx
4890 \DeclareMicrotypeAlias{zgmj}{ugm}    % "
4891 \DeclareMicrotypeAlias{zgmI}{ugm}    % "
4892 \DeclareMicrotypeAlias{zgmq}{ugm}    % "

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the config-
uration.
4893 \DeclareMicrotypeAlias{ulg} {blg}     % URW LetterGothic -> Bitstream LetterGothic12Pitch

Euro symbol fonts, to save some files.
4894 \DeclareMicrotypeAlias{zpeus} {zpeu}  % Adobe Euro sans -> serif
4895 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
4896 \DeclareMicrotypeAlias{euroitcs}{euroitc}% ITC Euro sans -> serif
4897

```

### 15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```

4898 %%% -----
4899 %%% INTERACTION WITH THE `babel' PACKAGE
4900
4901 \DeclareMicrotypeBabelHook
4902   {english,UKenglish,british,USenglish,american}
4903   {kerning=, spacing=nonfrench}
4904
4905 \DeclareMicrotypeBabelHook
4906   {french,français,acadian,canadien}
4907   {kerning=french, spacing=}
4908
4909 \DeclareMicrotypeBabelHook
4910   {turkish}
4911   {kerning=turkish, spacing=}
4912

```

### 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  $\LaTeX$  way, that is, when they have been assigned a slot in the font encoding with `\DeclareTextSymbol` or `\DeclareTextComposite`. Characters defined via `\chardef` are also possible.

Ligatures and `\mathchardef` symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed, or else in combination with the ‘inputenc’ key.

With  $X_{\LaTeX}$  or  $\text{Lua}\TeX$ , in contrast, it is advisable to use the proper Unicode characters.

### 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.

```

4913 </m-t>
4914 <*m-t|zpeu|mys>
4915 %%% -----
4916 %%% CHARACTER INHERITANCE
4917

```

```
4918 </m-t|zpeu|mys>
4919 <*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.

```
4920 \DeclareCharacterInheritance
4921   { encoding = OT1 }
4922   { f = {011}, % ff
4923     i = {\i},
4924     j = {\j},
4925     O = {\O},
4926     o = {\o}
4927   }
4928
```

### 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.

```
4929 \DeclareCharacterInheritance
4930   { encoding = T1 }
4931   { A = {\^A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
4932     a = {\^a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
4933     C = {\'C,\c C,\v C},
4934     c = {\'c,\c c,\v c},
4935     D = {\v D,\DH},
4936     d = {\v d,\dj},
4937     E = {\^E,\'E,\^E,\^E,\"E,\k E,\v E},
4938     e = {\^e,\'e,\^e,\^e,\"e,\k e,\v e},
4939     f = {027}, % ff
4940     G = {\u G},
4941     g = {\u g},
4942     I = {\^I,\'I,\^I,\^I,\"I,\.I},
4943     i = {\^i,\'i,\^i,\^i,\"i,\i},
4944     j = {\j},
4945     L = {\L,\'L,\v L},
4946     l = {\l,\'l,\v l},
4947     N = {\'N,\-N,\v N},
4948     n = {\'n,\-n,\v n},
4949     O = {\O,\^O,\'O,\^O,\-O,\"O,\H O},
4950     o = {\o,\^o,\'o,\^o,\-o,\"o,\H o},
4951     R = {\'R,\v R},
4952     r = {\'r,\v r},
4953     S = {\'S,\c S,\v S,\SS},
4954     s = {\'s,\c s,\v s},
4955     T = {\c T,\v T},
4956     t = {\c t,\v t},
4957     U = {\^U,\'U,\^U,\"U,\H U,\r U},
4958     u = {\^u,\'u,\^u,\"u,\H u,\r u},
4959     Y = {\'Y,\"Y},
4960     y = {\'y,\"y},
4961     Z = {\'Z,\.Z,\v Z},
4962     z = {\'z,\.z,\v z}
```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```
4963 % - = {127},
4964 }
4965
```

**15.5.3 LY1**

More characters: 008 ('fl'), 012 ('fi'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4966 \DeclareCharacterInheritance
4967   { encoding = LY1 }
4968   { A = {\^A,\'A,\^A,\-A,\"A,\r A},
4969     a = {\^a,\'a,\^a,\-a,\"a,\r a},
4970     C = {\c C},
4971     c = {\c c},
4972     D = {\DH},
4973     E = {\^E,\'E,\^E,\"E},
4974     e = {\^e,\'e,\^e,\"e},
4975     f = {011}, % ff
4976     I = {\^I,\'I,\^I,\"I},
4977     i = {\^i,\'i,\^i,\"i,\i},
4978     L = {\L},
4979     l = {\l},
4980     N = {\-N},
4981     n = {\-n},
4982     O = {\^O,\'O,\^O,\-O,\"O,\O},
4983     o = {\^o,\'o,\^o,\-o,\"o,\o},
4984     S = {\v S},
4985     s = {\v s},
4986     U = {\^U,\'U,\^U,\"U},
4987     u = {\^u,\'u,\^u,\"u},
4988     Y = {\'Y,\"Y},
4989     y = {\'y,\"y},
4990     Z = {\v Z},
4991     z = {\v z}
4992   }
4993

```

**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.

```

4994 \DeclareCharacterInheritance
4995   { encoding = OT4 }
4996   { A = {\k A},
4997     a = {\k a},
4998     C = {\'C},
4999     c = {\'c},
5000     E = {\k E},
5001     e = {\k e},
5002     f = {011}, % ff
5003     i = {\i},
5004     j = {\j},
5005     L = {\L},
5006     l = {\l},
5007     N = {\'N},
5008     n = {\'n},
5009     O = {\O,\"O},
5010     o = {\o,\"o},
5011     S = {\'S},
5012     s = {\'s},
5013     Z = {\'Z,\"Z},
5014     z = {\'z,\"z}
5015   }
5016

```

**15.5.5 QX**

The Central European QX encoding.<sup>16</sup> Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

5017 \DeclareCharacterInheritance
5018   { encoding = QX }
5019   { A = {\^A,\'A,\^A,\-A,\"A,\k A,\AA},
5020     a = {\`a,\'a,\^a,\-a,\"a,\k a,\aa},
5021     C = {\'C,\c C},
5022     c = {\'c,\c c},
5023     D = {\DH},
5024     E = {\^E,\'E,\^E,\"E,\k E},
5025     e = {\`e,\'e,\^e,\"e,\k e},
5026     f = {011}, % ff
5027     I = {\^I,\'I,\^I,\"I,\k I},
5028     i = {\`i,\'i,\^i,\"i,\k i,\i},
5029     j = {\j},
5030     L = {\L},
5031     l = {\l},
5032     N = {\'N,\-N},
5033     n = {\'n,\-n},
5034     O = {\0,\`0,\'0,\^0,\-0,\"0},
5035     o = {\o,\`o,\'o,\^o,\-o,\"o},

```

The Rumanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously<sup>17</sup>) been included in QX encoding. They are still kept for backwards compatibility.

```

5036   S = {\'S,\c S,\textcommabelow S,\v S},
5037   s = {\'s,\c s,\textcommabelow s,\v s},
5038   T = {\c T,\textcommabelow T},
5039   t = {\c t,\textcommabelow t},
5040   U = {\^U,\'U,\^U,\"U,\k U},
5041   u = {\`u,\'u,\^u,\"u,\k u},
5042   Y = {\'Y,\"Y},
5043   y = {\'y,\"y},
5044   Z = {\'Z,\-Z,\v Z},
5045   z = {\'z,\-z,\v z},
5046   . = \textellipsis
5047 }
5048

```

**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.

```

5049 \DeclareCharacterInheritance
5050   { encoding = T5 }
5051   { A = {\^A,\'A,\-A,\h A,\d A,\^A,\u A,
5052         \^{\Acircumflex},\'\Acircumflex,\-\Acircumflex,\h\Acircumflex,\d\Acircumflex,
5053         \^{\Abreve},\'\Abreve,\-\Abreve,\h\Abreve,\d\Abreve},
5054     a = {\`a,\'a,\-a,\h a,\d a,\^a,\u a,
5055         \^{\acircumflex},\'\acircumflex,\-\acircumflex,\h\acircumflex,\d\acircumflex,
5056         \^{\abreve},\'\abreve,\-\abreve,\h\abreve,\d\abreve},
5057     D = {\DJ},
5058     d = {\dj},
5059     E = {\^E,\'E,\-E,\h E,\d E,\^E,
5060         \^{\Ecircumflex},\'\Ecircumflex,\-\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
5061     e = {\`e,\'e,\-e,\h e,\d e,\^e,
5062         \^{\ecircumflex},\'\ecircumflex,\-\ecircumflex,\h\ecircumflex,\d\ecircumflex},

```

<sup>16</sup> Contributed by *Maciej Eder*.

<sup>17</sup> Cf. <http://tug.org/pipermail/tex-live/2008-August/017204.html>

```

5063 I = {\^I,\'I,\-I,\h I,\d I},
5064 i = {\^i,\'i,\-i,\h i,\d i,\i},
5065 O = {\^O,\'O,\-O,\h O,\d O,\^O,\horn O,
5066   \^Ocircumflex,\'Ocircumflex,\-Ocircumflex,\hOcircumflex,\dOcircumflex,
5067   \^Ohorn,\'Ohorn,\-Ohorn,\hOhorn,\dOhorn},
5068 o = {\^o,\'o,\-o,\h o,\d o,\^o,\horn o,
5069   \^ocircumflex,\'ocircumflex,\-ocircumflex,\hocircumflex,\docircumflex,
5070   \^ohorn,\'ohorn,\-ohorn,\hohorn,\dohorn},
5071 U = {\^U,\'U,\-U,\h U,\d U,\horn U,
5072   \^Uhorn,\'Uhorn,\-Uhorn,\hUhorn,\dUhorn},
5073 u = {\^u,\'u,\-u,\h u,\d u,\horn u,
5074   \^uhorn,\'uhorn,\-uhorn,\huhorn,\duhorn},
5075 Y = {\^Y,\'Y,\-Y,\h Y,\d Y},
5076 y = {\^y,\'y,\-y,\h y,\d y}
5077 }
5078

```

### 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.

```

5079 \DeclareCharacterInheritance
5080 { encoding = {EU1,EU2,TU} }
5081 { A = {\^A,\'A,\^A,\-A,\"A,\r A,\k A,\u A},
5082   a = {\^a,\'a,\^a,\-a,\"a,\r a,\k a,\u a},
5083   C = {\'C,\c C,\v C},
5084   c = {\'c,\c c,\v c},
5085   D = {\v D,\DH},
5086   d = {\v d,\dj},
5087   E = {\^E,\'E,\^E,\^E,\k E,\v E},
5088   e = {\^e,\'e,\^e,\^e,\k e,\v e},
5089 %   f = {/f_f}, % sometimes /f_f, sometimes /ff
5090   G = {\u G},
5091   g = {\u g},
5092   I = {\^I,\'I,\^I,\^I,\"I,\.I},
5093   i = {\^i,\'i,\^i,\^i,\"i,\.i},
5094 %   j = {\j},
5095   L = {\L,\'L,\v L},
5096   l = {\l,\'l,\v l},
5097   N = {\'N,\-N,\v N},
5098   n = {\'n,\-n,\v n},
5099   O = {\^O,\'O,\^O,\^O,\-O,\"O,\H O},
5100   o = {\^o,\'o,\^o,\^o,\-o,\"o,\H o},
5101   R = {\'R,\v R},
5102   r = {\'r,\v r},
5103   S = {\'S,\c S,\v S}, % \SS
5104   s = {\'s,\c s,\v s},
5105   T = {\c T,\v T},
5106   t = {\c t,\v t},
5107   U = {\^U,\'U,\^U,\^U,\H U,\r U},
5108   u = {\^u,\'u,\^u,\^u,\H u,\r u},
5109   Y = {\'Y,\"Y},
5110   y = {\'y,\"y},
5111   Z = {\'Z,\.Z,\v Z},
5112   z = {\'z,\.z,\v z}
5113 }
5114
5115 /m-t

```

### 15.5.8 Euro symbols

Make Euro symbols settings simpler.

```
5116 < *zpeu >
5117 \DeclareCharacterInheritance
5118   { encoding = U,
5119     family   = {zpeu,zpeus,eurosans} }
5120   { E = 128 }
5121
5122 < /zpeu >
5123 < *mvs >
```

Since 2006/05/11 (that is, one week after I've added these settings, after the package had been dormant for six years!), marvosym's encoding is (correctly) U instead of OT1.

```
5124 \DeclareCharacterInheritance
5125   { encoding = {OT1,U},
5126     family   = mvs }
5127   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
5128
5129 < /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.

```
5130 < *m-t >
5131 %%% -----
5132 %%% TRACKING/LETTERSPPACING
5133
5134 \SetTracking
5135   [ name       = default,
5136     no ligatures = {f} ]
5137   { encoding   = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
5138   { }
5139
```

### 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).

```
5140 %%% -----
5141 %%% EXPANSION
5142
5143 \SetExpansion
5144   [ name       = default      ]
5145   { encoding   = {OT1,OT4,QX,T1,LY1} }
5146   {
5147     A = 500,    a = 700,
5148     \AE = 500, \ae = 700,
5149     B = 700,    b = 700,
5150     C = 700,    c = 700,
5151     D = 500,    d = 700,
5152     E = 700,    e = 700,
5153     F = 700,
5154     G = 500,    g = 700,
5155     H = 700,    h = 700,
5156     K = 700,    k = 700,
5157     M = 700,    m = 700,
5158     N = 700,    n = 700,
5159     O = 500,    o = 700,
```

```

5160  \OE = 500,  \oe = 700,
5161  P = 700,   p = 700,
5162  Q = 500,   q = 700,
5163  R = 700,
5164  S = 700,   s = 700,
5165  U = 700,   u = 700,
5166  W = 700,   w = 700,
5167  Z = 700,   z = 700,
5168  2 = 700,
5169  3 = 700,
5170  6 = 700,
5171  8 = 700,
5172  9 = 700
5173  }
5174

```

### Settings for Cyrillic T2A encoding.<sup>18</sup>

```

5175 \SetExpansion
5176  [ name      = T2A ]
5177  { encoding = T2A }
5178  {
5179    A = 500,    a = 700,
5180    B = 700,    b = 700,
5181    C = 700,    c = 700,
5182    D = 500,    d = 700,
5183    E = 700,    e = 700,
5184    F = 700,
5185    G = 500,    g = 700,
5186    H = 700,    h = 700,
5187    K = 700,    k = 700,
5188    M = 700,    m = 700,
5189    N = 700,    n = 700,
5190    O = 500,    o = 700,
5191    P = 700,    p = 700,
5192    Q = 500,    q = 700,
5193    R = 700,
5194    S = 700,    s = 700,
5195    U = 700,    u = 700,
5196    W = 700,    w = 700,
5197    Z = 700,    z = 700,
5198    2 = 700,
5199    3 = 700,
5200    6 = 700,
5201    8 = 700,
5202    9 = 700,
5203    \CYRA = 500,  \cyra = 700,
5204    \CYRB = 700,  \cyrb = 700,
5205    \CYRV = 700,  \cyrv = 700,
5206    \CYRG = 700,  \cyrg = 700,
5207    \CYRD = 700,  \cyrd = 700,
5208    \CYRE = 700,  \cyre = 700,
5209    \CYRZH = 700, \cyrzh = 700,
5210    \CYRZ = 700,  \cyrz = 700,
5211    \CYRI = 700,  \cyri = 700,
5212    \CYRISHRT = 700, \cyrishrt = 700,
5213    \CYRK = 700,  \cyrk = 700,
5214    \CYRL = 700,  \cyrl = 700,
5215    \CYRM = 700,  \cyrm = 700,
5216    \CYRN = 700,  \cyrn = 700,
5217    \CYRO = 500,  \cyro = 700,
5218    \CYRP = 700,  \cyrp = 700,
5219    \CYRR = 700,  \cyrr = 700,
5220    \CYRS = 700,  \cyrs = 700,
5221    \CYRT = 700,  \cyrt = 700,

```

```

5222 \CYRU = 700, \cyru = 700,
5223 \CYRF = 700, \cyrf = 700,
5224 \CYRH = 700, \cyrh = 700,
5225 \CYRC = 700, \cyrc = 700,
5226 \CYRCH = 700, \cyrch = 700,
5227 \CYRSH = 700, \cyrsh = 700,
5228 \CYRSHCH = 700, \cyrshch = 700,
5229 \CYRHRDSN = 700, \cyrhrdsn = 700,
5230 \CYRERY = 700, \cyrery = 700,
5231 \CYRSFTSN = 700, \cyrsoftsn = 700,
5232 \CYREREV = 700, \cyrerev = 700,
5233 \CYRYU = 700, \cyryu = 700,
5234 \CYRYA = 700, \cyrya = 700
5235 }
5236

```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```

5237 \SetExpansion
5238 [ name = T5 ]
5239 { encoding = T5 }
5240 {
5241 A = 500, a = 700,
5242 B = 700, b = 700,
5243 C = 700, c = 700,
5244 D = 500, d = 700,
5245 E = 700, e = 700,
5246 F = 700,
5247 G = 500, g = 700,
5248 H = 700, h = 700,
5249 K = 700, k = 700,
5250 M = 700, m = 700,
5251 N = 700, n = 700,
5252 O = 500, o = 700,
5253 P = 700, p = 700,
5254 Q = 500, q = 700,
5255 R = 700,
5256 S = 700, s = 700,
5257 U = 700, u = 700,
5258 W = 700, w = 700,
5259 Z = 700, z = 700,
5260 2 = 700,
5261 3 = 700,
5262 6 = 700,
5263 8 = 700,
5264 9 = 700
5265 }
5266
5267 </m-t>

```

## 15.8 Character protrusion

```

5268 %%% -----
5269 %%% PROTRUSION
5270

```

For future historians, Hàn Thế Thành's original settings (from protcode.tex, converted to mi crotpe 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.

```

5271 <*cfg-t>
5272 \SetProtrusion
5273 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```
5274 <bch> [ name = bch-default ]
```

- Bitstream Letter Gothic (blg)

```
5275 <blg> [ name = blg-default ]
```

- Computer Modern Roman (cmr)

```
5276 <cmr> [ name = cmr-default ]
```

- Adobe Garamond (pad, padx, padj)

```
5277 <pad> [ name = pad-default ]
```

- Minion<sup>19</sup> (pmnx, pmnj)

```
5278 <pmn> [ name = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
5279 <ppl> [ name = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
5280 <ptm> [ name = ptm-default ]
```

- URW Garamond (ugm)

---

19 Contributed by *Harald Harders* and *Karl Karlsson*.

```

5281 <ugm> [ name = ugm-default ]
5282 <m-t|cmr|pmn> { }
5283 <bch|blg|pad|ugm> { encoding = OT1,
5284 <ppl|ptm> { encoding = {OT1,OT4},
5285 <bch> family = bch }
5286 <blg> family = blg }
5287 <pad> family = {pad,padx,padj} }
5288 <ppl> family = {ppl,pplx,pplj} }
5289 <ptm> family = {ptm,ptmx,ptmj} }
5290 <ugm> family = ugm }
5291 {
5292 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> A = {50,50},
5293 <ugm> A = {50,100},
5294 <pad|ptm> \AE = {50, },
5295 <ugm> \AE = {150,50},
5296 <ugm> B = { ,50},
5297 <bch|pad|pmn|ugm> C = {50, },
5298 <bch|pad|pmn> D = { ,50},
5299 <ugm> D = { ,70},
5300 <ugm> E = { ,50},
5301 <m-t|bch|cmr|pad|pmn|ptm> F = { ,50},
5302 <ugm> F = { ,70},
5303 <bch|pad|pmn> G = {50, },
5304 <ugm> G = {50,50},
5305 <blg> I = {150,150},
5306 <m-t|cmr|pad|pmn|ppl|ptm|ugm> J = {50, },
5307 <bch|blg> J = {100, },
5308 <!blg> K = { ,50},
5309 <blg> K = {50, },
5310 <m-t|bch|cmr|pad|pmn|ppl> L = { ,50},
5311 <blg> L = { ,150},
5312 <ptm> L = { ,80},
5313 <ugm> L = { ,120},
5314 <bch|pad|pmn|ugm> O = {50,50},
5315 <pad> \OE = {50, },
5316 <ugm> \OE = {50,50},
5317 <blg> P = { ,100},
5318 <ugm> P = { ,50},
5319 <bch|pad|pmn> Q = {50,70},
5320 <ugm> Q = {50,50},
5321 <bch> R = { ,50},
5322 <ugm> R = { ,70},
5323 <m-t|bch|cmr|pad|pmn|ppl|ptm> T = {50,50},
5324 <blg> T = {100,100},
5325 <ugm> T = {70,70},
5326 <m-t|bch|cmr|pad|pmn|ppl|ptm> V = {50,50},
5327 <blg|ugm> V = {70,70},
5328 <m-t|bch|cmr|pad|pmn|ppl|ptm> W = {50,50},
5329 <ugm> W = {70,70},
5330 <m-t|bch|cmr|pad|pmn|ppl|ptm> X = {50,50},
5331 <ugm> X = {50,70},
5332 <m-t|bch|cmr|pad|pmn|ppl> Y = {50,50},
5333 <blg|ptm|ugm> Y = {80,80},
5334 <ugm> Z = {50,50},
5335 <blg> f = {150,100},
5336 <blg> i = {150,150},
5337 <blg> j = {100,100},
5338 <m-t|bch|cmr|pad|pmn|ppl|ptm> k = { ,50},
5339 <ugm> k = { ,70},
5340 <blg> l = {150,150},
5341 <pmn> l = { , -50},
5342 <pad|ppl> p = {50,50},
5343 <ugm> p = { ,50},
5344 <pad|ppl> q = {50, },
5345 <!blg> r = { ,50},

```

```

5346 <blg> r = {100, 80},
5347 <cmr|pad|pmn> t = { ,70},
5348 <bch> t = { ,50},
5349 <blg> t = {150, 80},
5350 <ugm> t = { ,100},
5351 <m-t|bch|cmr|pad|pmn|ppl|ptm> v = {50,50},
5352 <blg> v = {100,100},
5353 <ugm> v = {50,70},
5354 <m-t|bch|cmr|pad|pmn|ppl|ptm> w = {50,50},
5355 <ugm> w = {50,70},
5356 <!blg> x = {50,50},
5357 <blg> x = {100,100},
5358 <m-t|bch|pad|pmn> y = { ,50},
5359 <blg> y = { 50,100},
5360 <cmr|ppl|ptm> y = {50,70},
5361 <ugm> y = { ,70},

5362 <cmr> 0 = { ,50},
5363 <m-t> 1 = {50,50},
5364 <bch|blg|pad|ptm|ugm> 1 = {150,150},
5365 <cmr> 1 = {100,200},
5366 <pmn> 1 = { ,50},
5367 <ppl> 1 = {100,100},
5368 <bch|cmr|pad|ugm> 2 = {50,50},
5369 <blg> 2 = { ,100},
5370 <bch|pmn> 3 = {50, },
5371 <cmr|pad|ugm> 3 = {50,50},
5372 <blg> 3 = {100, },
5373 <m-t|pad> 4 = {50,50},
5374 <bch> 4 = {100,50},
5375 <blg> 4 = {100, },
5376 <cmr|ugm> 4 = {70,70},
5377 <pmn> 4 = {50, },
5378 <ptm> 4 = {70, },
5379 <cmr> 5 = { ,50},
5380 <pad> 5 = {50,50},
5381 <bch> 6 = {50, },
5382 <cmr> 6 = { ,50},
5383 <pad> 6 = {50,50},
5384 <m-t> 7 = {50,50},
5385 <bch|pad|pmn|ugm> 7 = {50,80},
5386 <blg> 7 = {100,100},
5387 <cmr|ptm> 7 = {50,100},
5388 <ppl> 7 = { ,50},
5389 <cmr> 8 = { ,50},
5390 <bch|pad> 9 = {50,50},
5391 <cmr> 9 = { ,50},
5392 <m-t|cmr|pad|pmn|ppl|ptm|ugm> . = { ,700},
5393 <bch> . = { ,600},
5394 <blg> . = {400,500},
5395 <!blg> {,}= { ,500},
5396 <blg> {,}= {300,400},
5397 <m-t|cmr|pad|pmn|ppl|ptm|ugm> : = { ,500},
5398 <bch> : = { ,400},
5399 <blg> : = {300,400},
5400 <m-t|bch|pad|pmn|ptm> ; = { ,300},
5401 <blg> ; = {200,300},
5402 <cmr|ppl> ; = { ,500},
5403 <ugm> ; = { ,400},
5404 <!blg> ! = { ,100},
5405 <blg> ! = {200,200},
5406 <m-t|pad|pmn|ptm> ? = { ,100},
5407 <bch|cmr|ppl|ugm> ? = { ,200},
5408 <blg> ? = {150,150},
5409 <pmn> " = {300,300},
5410 <m-t|bch|cmr|pad|pmn|ppl> @ = {50,50},

```

```

5411 <ptm> @ = {100,100},
5412 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
5413 <ugm> ~ = {300,350},
5414 <pad|ppl|ptm> & = {50,100},
5415 <ugm> & = { ,100},
5416 <m-t|cmr|pad|pmn> \% = {50,50},
5417 <bch> \% = { ,50},
5418 <ppl|ptm> \% = {100,100},
5419 <ugm> \% = {50,100},
5420 <blg> \# = {100,100},
5421 <m-t|ppl|ptm|ugm> * = {200,200},
5422 <bch|pmn> * = {200,300},
5423 <blg> * = {150,200},
5424 <cmr|pad> * = {300,300},
5425 <m-t|cmr|ppl|ptm> + = {250,250},
5426 <bch> + = {150,250},
5427 <pad> + = {300,300},
5428 <blg|pmn> + = {150,200},
5429 <ugm> + = {250,300},
5430 <blg|ugm> {=} = {200,200},
5431 <m-t|pad|pmn|ptm> ( = {100, }, ) = { ,200},
5432 <bch|ugm> ( = {200, }, ) = { ,200},
5433 <cmr|blg> ( = {300, }, ) = { ,300},
5434 <ppl> ( = {100, }, ) = { ,300},
5435 <bch|pmn> [ = {100, }, ] = { ,100},
5436 <blg> [ = {300,100}, ] = { ,300},

5437 <m-t|pad|pmn|ptm> / = {100,200},
5438 <bch> / = { ,200},
5439 <blg> / = {300,300},
5440 <cmr|ppl> / = {200,300},
5441 <ugm> / = {100,300},
5442 <m-t|ptm> - = {500,500},
5443 <bch|cmr|ppl> - = {400,500},
5444 <blg> - = {300,400},
5445 <pad> - = {300,500},
5446 <pmn> - = {200,400},
5447 <ugm> - = {500,600},
5448 <blg> < = {200,100}, > = {100,200},
5449 <blg> _ = {150,250},
5450 <blg> | = {250,250},
5451 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
5452 <bch> \textendash = {200,300}, \textendash = {150,250},
5453 <cmr> \textendash = {400,300}, \textendash = {300,200},
5454 <pad|ppl|ptm> \textendash = {300,300}, \textendash = {200,200},
5455 <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).

```

5456 <m-t|bch|pmn> \textquoteleft = {300,400}, \textquoteright = {300,400},
5457 <blg> \textquoteleft = {400,600}, \textquoteright = {400,600},
5458 <cmr> \textquoteleft = {500,700}, \textquoteright = {500,600},
5459 <pad|ppl> \textquoteleft = {500,700}, \textquoteright = {500,700},
5460 <ptm> \textquoteleft = {500,500}, \textquoteright = {300,500},
5461 <ugm> \textquoteleft = {300,600}, \textquoteright = {300,600},
5462 <m-t|bch|pmn> \textquotedblleft = {300,300}, \textquotedblright = {300,300}
5463 <blg> \textquotedblright = {300,400}
5464 <cmr> \textquotedblleft = {500,300}, \textquotedblright = {200,600}
5465 <pad|ppl|ptm> \textquotedblleft = {300,400}, \textquotedblright = {300,400}
5466 <ugm> \textquotedblleft = {400,400}, \textquotedblright = {400,400}
5467 }
5468

```

Greek uppercase letters are in OT1 encoding only.

```

5469 <*m-t|cmr|pmn>

```

```

5470 \SetProtrusion
5471 <m-t> [ name = OT1-default,
5472 <cmr> [ name = cmr-OT1,
5473 <pmn> [ name = pmnj-OT1,
5474 <m-t> load = default ]
5475 <cmr> load = cmr-default ]
5476 <pmn> load = pmnj-default ]
5477 <m-t> { encoding = OT1 }
5478 <cmr> { encoding = {OT1,OT4},
5479 <pmn> { encoding = OT1,
5480 <cmr> family = cmr }
5481 <pmn> family = pmnj }
5482 {
5483 <m-t|cmr> \AE = {50, },
5484 <pmn> \OE = {50, }
5485 <*cmr>
5486 "00 = { ,150}, % \Gamma
5487 "01 = {100,100}, % \Delta
5488 "02 = { 50, 50}, % \Theta
5489 "03 = {100,100}, % \Lambda
5490 "06 = { 50, 50}, % \Sigma
5491 "07 = {100,100}, % \Upsilon
5492 "08 = { 50, 50}, % \Phi
5493 "09 = { 50, 50} % \Psi

```

Remaining slots can be found in the source file.

```

5494 </cmr>
5495 }
5496
5497 </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).

```

5498 \SetProtrusion
5499 <m-t> [ name = T1-default,
5500 <bch> [ name = bch-T1,
5501 <blg> [ name = blg-T1,
5502 <cmr> [ name = cmr-T1,
5503 <pad> [ name = pad-T1,
5504 <pmn> [ name = pmnj-T1,
5505 <ppl> [ name = ppl-T1,
5506 <ptm> [ name = ptm-T1,
5507 <ugm> [ name = ugm-T1,
5508 <m-t> load = default ]
5509 <bch> load = bch-default ]
5510 <blg> load = blg-default ]
5511 <cmr> load = cmr-default ]
5512 <pad> load = pad-default ]
5513 <pmn> load = pmnj-default ]
5514 <ppl> load = ppl-default ]
5515 <ptm> load = ptm-default ]
5516 <ugm> load = ugm-default ]
5517 <m-t> { encoding = {T1,LY1,EU1,EU2,TU} }
5518 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
5519 <blg|ptm|ugm> { encoding = {T1},
5520 <bch> family = bch }
5521 <blg> family = blg }
5522 <cmr> family = cmr }
5523 <pad> family = {pad,padx,padj} }
5524 <pmn> family = pmnj }
5525 <ppl> family = {ppl,pplx,pplj} }
5526 <ptm> family = {ptm,ptmx,ptmj} }
5527 <ugm> family = ugm }
5528 {

```

```

5529 <m-t|cmr> \AE = {50, },
5530 <bch|pmn> \OE = {50, },
5531 <pmn> \TH = { ,50},
5532 <blg> \v L = { ,250},
5533 <blg> \v d = { ,250},
5534 <blg> \v l = { ,250},
5535 <blg> \v t = { ,250},
5536 <blg> 127 = {300,400},
5537 <blg> 156 = {100, }, % IJ
5538 <blg> 188 = { 80, 80}, % ij
5539 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
5540 <cmr> _ = {200,200},
5541 <ugm> _ = {100,200},
5542 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
5543 <bch> \textbackslash = {150,200},
5544 <blg> \textbackslash = {250,300},
5545 <cmr|ppl> \textbackslash = {200,300},
5546 <ugm> \textbackslash = {100,300},
5547 <ugm> \textbar = {200,200},
5548 <blg> \textendash = {300,300}, \textemdash = {150,150},
5549 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
5550 <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.

```

5551 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5552 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
5553 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5554 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
5555 <blg> \guilsinglleft = {300,500}, \guilsingright = {300,500},
5556 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
5557 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
5558 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5559 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5560 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
5561 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5562 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
5563 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
5564 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
5565 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
5566 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
5567 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
5568 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
5569 <pmn> \textless = {100, }, \textgreater = { ,100},
5570 <pmn> \textvisiblespace = {100,100} % not in LY1

5571 }
5572

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

5573 <*cmr>
5574 \SetProtrusion
5575 [ name = lmr-T1,
5576 load = cmr-T1 ]
5577 { encoding = {T1,LY1},
5578 family = lmr }
5579 {
5580 \textquotedblleft = {300,400}, \textquotedblright = {300,400}
5581 }
5582
5583 </cmr>

```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).<sup>20</sup>

```

5584 <*m-t|cmr|pmn)
5585 \SetProtrusion
5586 <m-t) [ name = T2A-default,
5587 <cmr) [ name = cmr-T2A,
5588 <pmn) [ name = pmnj-T2A,
5589 <m-t) load = default ]
5590 <cmr) load = cmr-default ]
5591 <pmn) load = pmnj-default ]
5592 { encoding = T2A,
5593 <m-t) }
5594 <cmr) family = cmr }
5595 <pmn) family = pmnj }
5596 {
5597 \CYRA = {50,50},
5598 \CYRG = { ,50},
5599 \CYRK = { ,50},
5600 \CYRT = {50,50},
5601 \CYRH = {50,50},
5602 \CYRU = {50,50},
5603 <pmn) \CYRS = {50, },
5604 <pmn) \CYRO = {50,50},
5605 \cyrk = { ,50},
5606 \cyrg = { ,50},
5607 \cyrh = {50,50},
5608 <m-t|pmn) \cyru = {50,50},
5609 <cmr) \cyru = {50,70},
5610 <m-t) - = {100,100},
5611 <cmr) - = {200,200},
5612 <m-t) \textbackslash = {100,200}, \quotedblbase = {400,400},
5613 <cmr) \textbackslash = {200,300}, \quotedblbase = {400,400},
5614 <pmn) \textbackslash = {100,200}, \quotedblbase = {300,300},
5615 <cmr) \textquotedbl = {300,300}, \textquotedblleft = {200,600},
5616 <m-t) \guillemotleft = {200,200}, \guillemotright = {200,200},
5617 <cmr) \guillemotleft = {300,200}, \guillemotright = {100,400},
5618 <pmn) \guillemotleft = {200,200}, \guillemotright = {150,300},
5619 <m-t|cmr) \textbraceleft = {400,200}, \textbraceright = {200,400},
5620 <pmn) \textbraceleft = {200, }, \textbraceright = { ,300},
5621 <m-t|cmr) \textless = {200,100}, \textgreater = {100,200}
5622 <pmn) \textless = {100, }, \textgreater = { ,100}
5623 }
5624
5625 </m-t|cmr|pmn)

```

Settings for the QX encoding (generic and Times).<sup>21</sup> It also includes some glyphs otherwise in TS1.

```

5626 <*m-t|ptm)
5627 \SetProtrusion
5628 <m-t) [ name = QX-default,
5629 <ptm) [ name = ptm-QX,
5630 <m-t) load = default ]
5631 <ptm) load = ptm-default ]
5632 <m-t) { encoding = QX }
5633 <ptm) { encoding = QX,
5634 <ptm) family = {ptm,ptmx,ptmj} }
5635 {
5636 \AE = {50, },
5637 <ptm) * = {200,200},
5638 {=} = {100,100},
5639 \textunderscore = {100,100},
5640 \textbackslash = {100,200},
5641 <ptm) \quotedblbase = {400,400},

```

---

20 Contributed by Karl Karlsson.

21 Contributed by Maciej Eder.

```

5642 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
5643 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
5644 \textexclamdown = {100, }, \textquestiondown = {100, },
5645 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
5646 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
5647 \textless = {200,100}, \textgreater = {100,200},
5648 \textminus = {200,200}, \textdegree = {300,300},
5649 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5650 <ptm> \copyright = {100,150}, \textregistered = {100,150},
5651 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
5652 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
5653 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
5654 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
5655 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
5656 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5657 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5658 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
5659 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5660 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
5661 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5662 <ptm> \textperthousand = { ,50}
5663 }
5664
5665 </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.

```

5666 <*cmr|bch>
5667 \SetProtrusion
5668 <cmr> [ name = cmr-T5,
5669 <cmr> load = cmr-default ]
5670 <bch> [ name = bch-T5,
5671 <bch> load = bch-default ]
5672 { encoding = T5,
5673 <cmr> family = cmr }
5674 <bch> family = bch }
5675 {
5676 <bch> _ = {100,100},
5677 <bch> \textbackslash = {150,200},
5678 <cmr> \textbackslash = {200,300},
5679 <cmr> \textquotedblleft = {200,600},
5680 <cmr> \textquotedbl = {300,300},
5681 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
5682 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
5683 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
5684 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5685 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},
5686 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
5687 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
5688 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
5689 \textless = {200,100}, \textgreater = {100,200}
5690 }
5691
5692 </cmr|bch>

```

Minion with lining numbers.

```

5693 <*pmn>
5694 \SetProtrusion
5695 [ name = pmnx-OT1,
5696 load = pmnj-default ]
5697 { encoding = OT1,
5698 family = pmnx }
5699 {
5700 1 = {230,180}
5701 }

```

```

5702
5703 \SetProtrusion
5704 [ name = pmnx-T1,
5705   load = pmnj-T1 ]
5706 { encoding = {T1,LY1},
5707   family = pmnx }
5708 {
5709   1 = {230,180}
5710 }
5711
5712 \SetProtrusion
5713 [ name = pmnx-T2A,
5714   load = pmnj-T2A ]
5715 { encoding = {T2A},
5716   family = pmnx }
5717 {
5718   1 = {230,180}
5719 }
5720
5721 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

5722 <*ptm>
5723 \SetProtrusion
5724 [ name = ptm-LY1,
5725   load = ptm-T1 ]
5726 { encoding = LY1,
5727   family = {ptm,ptmx,ptmj} }
5728 {
5729   - = {100,100},
5730   \texttrademark = {100,100},
5731   \textregistered = {100,100},
5732   \textcopyright = {100,100},
5733   \textdegree = {300,300},
5734   \textminus = {200,200},
5735   \textellipsis = {150,200},
5736 % \texteuro = { , }, % ?
5737   \textcent = {100,100},
5738   \textquotesingle = {500,500},
5739   \textflorin = { 50, 70},
5740   \textdagger = {150,150},
5741   \textdaggerdbl = {100,100},
5742   \textperthousand = { , 50},
5743   \textbullet = {150,150},
5744   \textonesuperior = {100,100},
5745   \texttwosuperior = { 50, 50},
5746   \textthreesuperior = { 50, 50},
5747   \textperiodcentered = {300,300},
5748   \textplusminus = { 50, 80},
5749   \textmultiply = {100,100},
5750   \textdivide = { 50,150}

```

Remaining slots in the source file.

```

5751   }
5752 }
5753 </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>22</sup>

```

5754 \SetProtrusion
5755 <m-t> [ name = OT1-it ]
5756 <bch> [ name = bch-it ]
5757 <blg> [ name = blg-it,
5758 <blg> load = blg-default ]
5759 <cmr> [ name = cmr-it ]
5760 <pad> [ name = pad-it ]
5761 <pmn> [ name = pmnj-it ]
5762 <ppl> [ name = ppl-it ]
5763 <ptm> [ name = ptm-it ]
5764 <ugm> [ name = ugm-it ]
5765 <m-t|bch|blg|pad|ugm> { encoding = OT1,
5766 <ppl|ptm> { encoding = {OT1,OT4},
5767 <bch> family = bch,
5768 <blg> family = blg,
5769 <pad> family = {pad,padx,padj},
5770 <ppl> family = {ppl,pplx,pplj},
5771 <ptm> family = {ptm,ptmx,ptmj},
5772 <ugm> family = ugm,
5773 <m-t|bch|pad|ppl|ptm> shape = {it,sl} }
5774 <blg|ugm> shape = it }
5775 <cmr|pmn> { }
5776 {
5777 <cmr> A = {100,100},
5778 <ptm> A = {100,50},
5779 <pad|pmn> A = {50, },
5780 <ugm> A = { ,150},
5781 <ppl> A = {50,50},
5782 <ptm> \AE = {100, },
5783 <pad|ppl> \AE = {50, },
5784 <cmr> B = {83,-40},
5785 <pad|ppl|ptm> B = {50, },
5786 <pmn> B = {20,-50},
5787 <bch|ppl|ptm|ugm> C = {50, },
5788 <cmr> C = {165,-75},
5789 <pad> C = {100, },
5790 <pmn> C = {50,-50},
5791 <cmr> D = {75, -28},
5792 <pad|ppl|ptm> D = {50,50},
5793 <pmn> D = {20, },
5794 <cmr> E = {80,-55},
5795 <pad|ppl|ptm> E = {50, },
5796 <pmn> E = {20,-50},
5797 <cmr> F = {85,-80},
5798 <pad|ptm> F = {100, },
5799 <pmn> F = {10, },
5800 <ppl> F = {50, },
5801 <bch|ppl|ptm|ugm> G = {50, },
5802 <cmr> G = {153,-15},
5803 <pad> G = {100, },
5804 <pmn> G = {50,-50},
5805 <cmr> H = {73,-60},
5806 <pad|ppl|ptm> H = {50, },
5807 <cmr> I = {140,-120},
5808 <pad|ptm> I = {50, },
5809 <pmn> I = {20,-50},
5810 <cmr> J = {135,-80},
5811 <pad> J = {50, },
5812 <pmn> J = {20, },

```

22 Settings contributed by Hendrik Vogt.

```

5813 <ptm> J = {100, },
5814 <cmr> K = {70,-30},
5815 <pad|ppl|ptm> K = {50, },
5816 <pmn> K = {20, },
5817 <cmr> L = {87, 40},
5818 <pad|ppl|ptm> L = {50, },
5819 <pmn> L = {20,50},
5820 <ugm> L = { ,100},
5821 <cmr> M = {67,-45},
5822 <pmn> M = { , -30},
5823 <ptm> M = {50, },
5824 <cmr> N = {75,-55},
5825 <pmn> N = { , -30},
5826 <ptm> N = {50, },
5827 <bch|pmn|ppl|ptm> O = {50, },
5828 <cmr> O = {150,-30},
5829 <pad> O = {100, },
5830 <ugm> O = {70,50},
5831 <ppl|ptm> \OE = {50, },
5832 <pad> \OE = {100, },
5833 <cmr> P = {82,-50},
5834 <pad|ppl|ptm> P = {50, },
5835 <pmn> P = {20,-50},
5836 <bch|pmn|ppl|ptm> Q = {50, },
5837 <cmr> Q = {150,-30},
5838 <pad> Q = {100, },
5839 <ugm> Q = {70,50},
5840 <cmr> R = {75, 15},
5841 <pad|ppl|ptm> R = {50, },
5842 <pmn> R = {20, },
5843 <bch|pad|ppl|ptm> S = {50, },
5844 <cmr> S = {90,-65},
5845 <pmn> S = {20,-30},
5846 <bch|pad|ppl|ptm> $ = {50, },
5847 <cmr> $ = {100,-20},
5848 <pmn> $ = {20,-30},
5849 <bch|pmn|ugm> T = {70, },
5850 <cmr> T = {220,-85},
5851 <pad|ppl|ptm> T = {100, },
5852 <cmr> U = {230,-55},
5853 <pad|ppl|ptm> U = {50, },
5854 <pmn> U = {50,-50},
5855 <cmr> V = {260,-60},
5856 <pad|pmn|ugm> V = {100, },
5857 <ppl|ptm> V = {100,50},
5858 <cmr> W = {185,-55},
5859 <pad|pmn|ugm> W = {100, },
5860 <ppl> W = {50, },
5861 <ptm> W = {100,50},
5862 <cmr> X = {70,-30},
5863 <ppl|ptm> X = {50, },
5864 <cmr> Y = {250,-60},
5865 <pmn> Y = {50, },
5866 <ppl> Y = {100,50},
5867 <ptm> Y = {100, },
5868 <cmr> Z = {90,-60},
5869 <pmn> Z = { , -50},
5870 <cmr> a = {150,-10},
5871 <cmr> b = {170, },
5872 <cmr> c = {173,-10},
5873 <cmr> d = {150,-55},
5874 <pmn> d = { , -50},
5875 <cmr> e = {180, },
5876 <cmr> f = { , -250},
5877 <pad|pmn> f = { , -100},

```

```

5878 <cmr> g = {150,-10},
5879 <cmr> h = {100, },
5880 <cmr> i = {210, },
5881 <pmn> i = { , -30},
5882 <cmr> j = { , -40},
5883 <pmn> j = { , -30},
5884 <cmr> k = {110,-50},
5885 <cmr> l = {240,-110},
5886 <pmn> l = { , -100},
5887 <cmr> m = {80, },
5888 <cmr> n = {115, },
5889 <bch> o = {50,50},
5890 <cmr> o = {155, },
5891 <bch> p = { , 50},
5892 <pmn> p = {-50, },
5893 <bch> q = {50, },
5894 <cmr> q = {170,-40},
5895 <cmr> r = {155,-40},
5896 <pmn> r = { , 50},
5897 <cmr> s = {130, },
5898 <bch> t = { , 50},
5899 <cmr> t = {230,-10},
5900 <cmr> u = {120, },
5901 <cmr> v = {140,-25},
5902 <pmn|ugm> v = {50, },
5903 <bch> w = { , 50},
5904 <cmr> w = {98,-20},
5905 <pmn|ugm> w = {50, },
5906 <cmr> x = {65,-40},
5907 <bch> y = { , 50},
5908 <cmr> y = {130,-20},
5909 <cmr> z = {110,-80},
5910 <cmr> 0 = {170,-85},
5911 <bch|ptm> 1 = {150,100},
5912 <cmr> 1 = {230,110},
5913 <pad> 1 = {150, },
5914 <pmn> 1 = {50, },
5915 <ppl> 1 = {100, },
5916 <ugm> 1 = {150,150},
5917 <cmr> 2 = {130,-70},
5918 <pad|ppl|ptm> 2 = {50, },
5919 <pmn> 2 = {-50, },
5920 <bch> 3 = {50, },
5921 <cmr> 3 = {140,-70},
5922 <pmn> 3 = {-100, },
5923 <ptm> 3 = {100,50},
5924 <bch> 4 = {100, },
5925 <cmr> 4 = {130,80},
5926 <pad> 4 = {150, },
5927 <ppl|ptm> 4 = {50, },
5928 <cmr> 5 = {160, },
5929 <ptm> 5 = {50, },
5930 <bch> 6 = {50, },
5931 <cmr> 6 = {175,-30},
5932 <bch|pad|ptm> 7 = {100, },
5933 <cmr> 7 = {250,-150},
5934 <pmn> 7 = {20, },
5935 <ppl> 7 = {50, },
5936 <cmr> 8 = {130,-40},
5937 <cmr> 9 = {155,-80},
5938 <m-t|cmr|pad|pmn|ppl> . = { , 500},
5939 <big> . = {400,600},
5940 <bch|ptm|ugm> . = { , 700},
5941 <big> {,} = {300,500},
5942 <m-t|pad|pmn|ppl> {,} = { , 500},

```

```

5943 <cmr> {,}= { ,450},
5944 <bch|ugm> {,}= { ,600},
5945 <ptm> {,}= { ,700},
5946 <m-t|cmr|pad|ppl> := { ,300},
5947 <bch|ugm> := { ,400},
5948 <pmn> := { ,200},
5949 <ptm> := { ,500},
5950 <m-t|cmr|pad|ppl> ; = { ,300},
5951 <bch|ugm> ; = { ,400},
5952 <pmn> ; = { ,200},
5953 <ptm> ; = { ,500},
5954 <ptm> != { ,100},
5955 <bch> ? = { ,200},
5956 <ptm> ? = { ,100},
5957 <ppl> ? = { ,300},
5958 <pmn> " = {400,200},
5959 <m-t|pad|pmn|ppl|ptm> & = {50,50},
5960 <bch> & = { ,80},
5961 <cmr> & = {130,30},
5962 <ugm> & = {50,100},
5963 <m-t|pad|pmn> \% = {100, },
5964 <cmr> \% = {180,50},
5965 <bch> \% = {50,50},
5966 <ppl|ptm> \% = {100,100},
5967 <ugm> \% = {100,50},
5968 <m-t|pmn|ppl> * = {200,200},
5969 <bch> * = {300,200},
5970 <cmr> * = {380,20},
5971 <pad> * = {500,100},
5972 <ptm|ugm> * = {400,200},
5973 <m-t|pmn|ppl> + = {150,200},
5974 <cmr> + = {180,200},
5975 <bch|ugm> + = {250,250},
5976 <pad|ptm> + = {250,200},
5977 <m-t|pad|pmn|ppl> @ = {50,50},
5978 <bch> @ = {80,50},
5979 <cmr> @ = {180,10},
5980 <ptm> @ = {150,150},
5981 <m-t|bch|ugm> ~ = {150,150},
5982 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
5983 <ugm> {=} = {200,200},
5984 <m-t|bch|pad|pmn|ppl|ptm|ugm> ( = {200, }, ) = { ,200},
5985 <cmr> ( = {300, }, ) = { ,70},
5986 <m-t|pad|ppl|ptm|ugm> / = {100,200},
5987 <cmr> / = {100,100},
5988 <bch> / = { ,150},
5989 <pmn> / = {100,150},
5990 <m-t> - = {300,300},
5991 <bch|pad> - = {300,400},
5992 <pmn> - = {200,300},
5993 <cmr> - = {500,300},
5994 <ppl> - = {300,500},
5995 <ptm> - = {500,500},
5996 <ugm> - = {400,700},
5997 <blg> - = {0,300},
5998 <m-t|pmn> \textendash = {200,200}, \textendash = {150,150},
5999 <bch> \textendash = {200,300}, \textendash = {150,200},
6000 <cmr> \textendash = {500,300}, \textendash = {400,170},
6001 <pad|ppl|ptm|ugm> \textendash = {300,300}, \textendash = {200,200},
6002 <m-t|bch|pmn|ugm> \textquoteleft = {400,200}, \textquoteright = {400,200},
6003 <blg> \textquoteleft = {400,400}, \textquoteright = {400,400},
6004 <cmr> \textquoteleft = {800,200}, \textquoteright = {800,-20},
6005 <pad> \textquoteleft = {800,200}, \textquoteright = {800,200},
6006 <ppl> \textquoteleft = {700,400}, \textquoteright = {700,400},
6007 <ptm> \textquoteleft = {800,500}, \textquoteright = {800,500},

```

```

6008 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
6009 <blg> \textquotedblright = {300,300}
6010 <cmr> \textquotedblleft = {540,100}, \textquotedblright = {500,100}
6011 <pad> \textquotedblleft = {700,200}, \textquotedblright = {700,200}
6012 <ppl> \textquotedblleft = {500,300}, \textquotedblright = {500,300}
6013 <ptm> \textquotedblleft = {700,400}, \textquotedblright = {700,400}
6014 <ugm> \textquotedblleft = {600,200}, \textquotedblright = {600,200}
6015 }
6016
6017 <*cmr|pmn>
6018 \SetProtrusion
6019 <cmr> [ name = cmr-it-OT1,
6020 <pmn> [ name = pmnj-it-OT1,
6021 <cmr> [ load = cmr-it ]
6022 <pmn> [ load = pmnj-it ]
6023 <cmr> { encoding = {OT1,OT4},
6024 <pmn> { encoding = OT1,
6025 <cmr> family = cmr,
6026 <pmn> family = pmnj,
6027 <cmr> shape = it }
6028 <pmn> shape = {it,s1} }
6029 {
6030 <cmr> \AE = {100, },
6031 <pmn> \AE = { , -50},
6032 <cmr> \OE = {100, },
6033 <pmn> \OE = {50, }
6034 <*cmr>
6035 "00 = {200,150}, % \Gamma
6036 "01 = {150,100}, % \Delta
6037 "02 = {150, 50}, % \Theta
6038 "03 = {150, 50}, % \Lambda
6039 "04 = {100,100}, % \Xi
6040 "05 = {100,100}, % \Pi
6041 "06 = {100, 50}, % \Sigma
6042 "07 = {200,150}, % \Upsilon
6043 "08 = {150, 50}, % \Phi
6044 "09 = {150,100}, % \Psi
6045 "0A = { 50, 50} % \Omega
6046 </cmr>
6047 }
6048
6049 </cmr|pmn>
6050 \SetProtrusion
6051 <m-t> [ name = T1-it-default,
6052 <bch> [ name = bch-it-T1,
6053 <blg> [ name = blg-it-T1,
6054 <cmr> [ name = cmr-it-T1,
6055 <pad> [ name = pad-it-T1,
6056 <pmn> [ name = pmnj-it-T1,
6057 <ppl> [ name = ppl-it-T1,
6058 <ptm> [ name = ptm-it-T1,
6059 <ugm> [ name = ugm-it-T1,
6060 <m-t> [ load = OT1-it ]
6061 <bch> [ load = bch-it ]
6062 <blg> [ load = blg-T1 ]
6063 <cmr> [ load = cmr-it ]
6064 <pmn> [ load = pmnj-it ]
6065 <pad> [ load = pad-it ]
6066 <ppl> [ load = ppl-it ]
6067 <ptm> [ load = ptm-it ]
6068 <ugm> [ load = ugm-it ]
6069 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
6070 <blg|ptm|ugm> { encoding = T1,
6071 <bch> family = bch,
6072 <blg> family = blg,

```

```

6073 <cmr> family = cmr,
6074 <pmn> family = pmnj,
6075 <pad> family = {pad,padx,padj},
6076 <ppl> family = {ppl,pplx,pplj},
6077 <ptm> family = {ptm,ptmx,ptmj},
6078 <ugm> family = ugm,
6079 <m-t|bch|pad|pmn|ppl|ptm> shape = {it,sl} }
6080 <blg|cmr|ugm> shape = it }
6081 {
6082 <m-t|bch|pmn> _ = { ,100},
6083 <blg> _ = {0,300},
6084 <cmr|ugm> _ = {100,200},
6085 <pad|ppl|ptm> _ = {100,100},
6086 <blg> . = {400,600},
6087 <blg> {,}= {300,500},
6088 <cmr> \AE = {100, },
6089 <pmn> \AE = { , -50},
6090 <bch|pmn> \OE = { 50, },
6091 <cmr> \OE = {100, },
6092 <pmn> O31 = { , -100}, % ffi
6093 <cmr|ptm> 156 = {100, }, % IJ
6094 <pad> 156 = {50, }, % IJ
6095 <pmn> 156 = {20, }, % IJ
6096 <pmn> 188 = { , -30}, % ij
6097 <pmn> \v t = { ,100},
6098 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
6099 <cmr|ugm> \textbackslash = {300,300},
6100 <bch> \textbackslash = {150,150},
6101 <pmn> \textbackslash = {100,150},
6102 <ugm> \textbar = {200,200},
6103 <cmr> \textquotedblleft = {500,300},
6104 <blg> \textquotedleft = {400,400}, \textquoteright = {400,400},
6105 <blg> \textquotedbl = {300,300}, \textquotedblleft = {300,300},
6106 <blg> \textquotedblright = {300,300}, \quotedblbase = {200,600},
6107 <m-t|ptm> \quotesinglbase = {300,700}, \quotedblbase = {400,500},
6108 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6109 <bch|pmn> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6110 <pad|ppl> \quotesinglbase = {500,500}, \quotedblbase = {400,400},
6111 <ugm> \quotesinglbase = {300,700}, \quotedblbase = {300,500},
6112 <m-t|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
6113 <bch|pmn> \guilsinglleft = {300,400}, \guilsingright = {200,500},
6114 <cmr> \guilsinglleft = {500,300}, \guilsingright = {400,400},
6115 <pad> \guilsinglleft = {500,400}, \guilsingright = {300,500},
6116 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
6117 <m-t|ppl> \guillemotleft = {300,300}, \guillemotright = {300,300},
6118 <bch|pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6119 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6120 <pad> \guillemotleft = {300,300}, \guillemotright = {200,400},
6121 <ptm> \guillemotleft = {300,400}, \guillemotright = {200,400},
6122 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
6123 <m-t|pad|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {200, },
6124 <cmr|ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
6125 <pmn> \textexclamdown = {-50, }, \textquestiondown = {-50, },
6126 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
6127 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
6128 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
6129 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
6130 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
6131 <pmn> \textvisiblespace = {100,100}
6132 }
6133
6134 <*m-t|cmr|pmn>
6135 \SetProtrusion
6136 <m-t> [ name = T2A-it-default,
6137 <cmr> [ name = cmr-it-T2A,

```

```

6138 <pmn> [ name      = pmnj-it-T2A,
6139 <m-t>   load      = OT1-it  ]
6140 <cmr>   load      = cmr-it  ]
6141 <pmn>   load      = pmnj-it  ]
6142   { encoding = T2A,
6143 <cmr>   family   = cmr,
6144 <pmn>   family   = pmnj,
6145 <m-t|pmn> shape   = {it,s1} }
6146 <cmr>   shape   = it      }
6147   {
6148 <cmr>   \CYRA = {100,50},
6149 <pmn>   \CYRA = {50, },
6150 <cmr>   \CYRB = {50, },
6151 <cmr>   \CYRV = {50, },
6152 <pmn>   \CYRV = {20,-50},
6153 <cmr>   \CYRG = {100, },
6154 <pmn>   \CYRG = {10, },
6155 <cmr>   \CYRD = {50, },
6156 <cmr>   \CYRE = {50, },
6157 <pmn>   \CYRE = {20,-50},
6158 <cmr>   \CYRZH = {50, },
6159 <cmr>   \CYRZ = {50, },
6160 <pmn>   \CYRZ = {20,-50},
6161 <cmr>   \CYRI = {50, },
6162 <pmn>   \CYRI = { , -30},
6163 <cmr>   \CYRISHRT = {50, },
6164 <cmr>   \CYRK = {50, },
6165 <pmn>   \CYRK = {20, },
6166 <cmr>   \CYRL = {50, },
6167 <cmr>   \CYRM = {50, },
6168 <pmn>   \CYRM = { , -30},
6169 <cmr>   \CYRN = {50, },
6170 <cmr>   \CYRO = {100, },
6171 <pmn>   \CYRO = {50, },
6172 <cmr>   \CYRP = {50, },
6173 <cmr>   \CYRR = {50, },
6174 <pmn>   \CYRR = {20,-50},
6175 <cmr>   \CYRS = {100, },
6176 <pmn>   \CYRS = {50, },
6177 <cmr>   \CYRT = {100, },
6178 <pmn>   \CYRT = {70, },
6179 <cmr>   \CYRU = {100, },
6180 <pmn>   \CYRU = {50, },
6181 <cmr>   \CYRF = {100, },
6182 <cmr>   \CYRH = {50, },
6183 <cmr>   \CYRC = {50, },
6184 <cmr>   \CYRCH = {100, },
6185 <cmr>   \CYRSH = {50, },
6186 <cmr>   \CYRSHCH = {50, },
6187 <cmr>   \CYRHRDSN = {100, },
6188 <cmr>   \CYRERY = {50, },
6189 <cmr>   \CYRSFTSN = {50, },
6190 <cmr>   \CYREREV = {50, },
6191 <cmr>   \CYRYU = {50, },
6192 <cmr>   \CYRYA = {50, },
6193 <pmn>   \CYRYA = { , 20},
6194 <pmn>   \cyrr = {-50, },
6195 <m-t|pmn>   _ = { , 100},
6196 <cmr>   _ = {100,200},
6197 <pmn>   031 = { , -100}, % ff1
6198 <pmn>   \v t = { , 100},
6199 <m-t>   \textbackslash = {100,200}, \quotedblbase = {400,500},
6200 <cmr>   \textbackslash = {300,300}, \quotedblbase = {200,600},
6201 <pmn>   \textbackslash = {100,150}, \quotedblbase = {150,500},
6202 <m-t>   \guillemotleft = {300,300}, \guillemotright = {300,300},

```

```

6203 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6204 <pmn> \guillemotleft = {200,300}, \guillemotright = {150,400},
6205 <m-t> \textbraceleft = {200,100}, \textbraceright = {200,200},
6206 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
6207 <pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
6208 <cmr> \textquotedblleft = {500,300},
6209 <cmr> \textless = {300,100}, \textgreater = {200,100}
6210 <pmn> \textless = {100, }, \textgreater = { ,100}
6211 }
6212
6213 </m-t|cmr|pmn>
6214 < *m-t|ptm>
6215 \SetProtrusion
6216 <m-t> [ name = QX-it-default,
6217 <ptm> [ name = ptm-it-QX,
6218 <m-t> load = OT1-it ]
6219 <ptm> load = ptm-it ]
6220 { encoding = {QX},
6221 <ptm> family = {ptm,ptmx,ptmj},
6222 shape = {it,s1} }
6223 {
6224 <ptm> 009 = { , 50}, % fk
6225 {=} = {100,100},
6226 <m-t> \textunderscore = {100,100},
6227 <ptm> \textunderscore = {100,150},
6228 \textbackslash = {100,200},
6229 \quotedblbase = {300,400},
6230 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
6231 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
6232 \textexclamdown = {200, }, \textquestiondown = {200, },
6233 \textbraceleft = {200,100}, \textbraceright = {200,200},
6234 \textless = {100,100}, \textgreater = {100,100},
6235 \textminus = {200,200}, \textdegree = {300,150},
6236 <m-t> \copyright = {100,100}, \textregistered = {100,100}
6237 <ptm> \textregistered = {100,150}, \copyright = {100,150},
6238 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
6239 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
6240 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
6241 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
6242 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
6243 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
6244 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
6245 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
6246 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
6247 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
6248 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
6249 <ptm> \textperthousand = { ,50}
6250 }
6251
6252 </m-t|ptm>
6253 < *cmr|bch>
6254 \SetProtrusion
6255 <cmr> [ name = cmr-it-T5,
6256 <cmr> load = cmr-it ]
6257 <bch> [ name = bch-it-T5,
6258 <bch> load = bch-it ]
6259 { encoding = T5,
6260 <bch> family = bch,
6261 <cmr> family = cmr,
6262 shape = it }
6263 {
6264 <bch> _ = { ,100},
6265 <cmr> _ = {100,200},
6266 <bch> \textbackslash = {150,150},
6267 <cmr> \textbackslash = {300,300},

```

```

6268 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},
6269 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
6270 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
6271 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
6272 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
6273 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
6274 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
6275 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
6276 <bch> \textless = {100, }, \textgreater = { ,100},
6277 <cmr> \textless = {300,100}, \textgreater = {200,100}
6278 }
6279
6280 </cmr|bch>

```

Slanted is very similar to italic.

```

6281 <*cmr>
6282 \SetProtrusion
6283 [ name = cmr-sl,
6284 load = cmr-it-OT1 ]
6285 { encoding = {OT1,OT4},
6286 family = cmr,
6287 shape = sl }
6288 {
6289 L = { ,50},
6290 f = { ,-50},
6291 - = {300, },
6292 \textendash = {400, }, \textemdash = {300, }
6293 }
6294
6295 \SetProtrusion
6296 [ name = cmr-sl-T1,
6297 load = cmr-it-T1 ]
6298 { encoding = {T1,LY1},
6299 family = cmr,
6300 shape = sl }
6301 {
6302 L = { ,50},
6303 f = { ,-50},
6304 - = {300, },
6305 \textendash = {400, }, \textemdash = {300, }
6306 }
6307
6308 \SetProtrusion
6309 [ name = cmr-sl-T2A,
6310 load = cmr-it-T2A ]
6311 { encoding = T2A,
6312 family = cmr,
6313 shape = sl }
6314 {
6315 L = { ,50},
6316 f = { ,-50},
6317 - = {300, },
6318 \textendash = {400, }, \textemdash = {300, }
6319 }
6320
6321 \SetProtrusion
6322 [ name = cmr-sl-T5,
6323 load = cmr-it-T5 ]
6324 { encoding = T5,
6325 family = cmr,
6326 shape = sl }
6327 {
6328 L = { ,50},
6329 f = { ,-50},
6330 - = {300, },

```

```

6331     \textendash = {400, }, \textemdash = {300, }
6332   }
6333
6334 \SetProtrusion
6335   [ name = lmr-it-T1,
6336     load = cmr-it-T1 ]
6337   { encoding = {T1,LY1},
6338     family = lmr,
6339     shape = {it,s1} }
6340   {
6341     \textquotedblleft = { ,200}, \textquotedblright = { ,200},
6342     \quotesinglbase = { ,400}, \quotedblbase = { ,500}
6343   }
6344

```

Oldstyle numerals are slightly different.

```

6345 \SetProtrusion
6346   [ name = cmr(oldstyle)-it,
6347     load = cmr-it-T1 ]
6348   { encoding = T1,
6349     family = {hfor,cmor},
6350     shape = {it,s1} }
6351   {
6352     1 = {250, 50},
6353     2 = {150,-100},
6354     3 = {100,-50},
6355     4 = {150,150},
6356     6 = {200, },
6357     7 = {200, 50},
6358     8 = {150,-50},
6359     9 = {100, 50}
6360   }
6361
6362 </cmr>
6363 < *pmn >
6364 \SetProtrusion
6365   [ name = pmnx-it,
6366     load = pmnj-it ]
6367   { encoding = OT1,
6368     family = pmnx,
6369     shape = {it,s1} }
6370   {
6371     1 = {100,150}
6372   }
6373
6374 \SetProtrusion
6375   [ name = pmnx-it-T1,
6376     load = pmnj-it-T1 ]
6377   { encoding = {T1,LY1},
6378     family = pmnx,
6379     shape = {it,s1} }
6380   {
6381     1 = {100,150}
6382   }
6383
6384 \SetProtrusion
6385   [ name = pmnx-it-T2A,
6386     load = pmnj-it-T2A ]
6387   { encoding = {T2A},
6388     family = pmnx,
6389     shape = {it,s1} }
6390   {
6391     1 = {100,150}
6392   }
6393

```

```

6394 </pmn>
6395 <*ptm>
6396 \SetProtrusion
6397 [ name = ptm-it-LY1,
6398   load = ptm-it-T1 ]
6399 { encoding = {LY1},
6400   family = {ptm,ptmx,ptmj},
6401   shape = {it,s1} }
6402 {
6403   – = {100,100},
6404   \texttrademark = {100,100},
6405   \textregistered = {100,100},
6406   \textcopyright = {100,100},
6407   \textdegree = {300,100},
6408   \textminus = {200,200},
6409   \textellipsis = {100,200},
6410 % \texteuro = { , }, % ?
6411   \textcent = {100,100},
6412   \textquotesingle = {500, },
6413   \textflorin = {100, 70},
6414   \textdagger = {150,150},
6415   \textdaggerdbl = {100,100},
6416   \textbullet = {150,150},
6417   \textonesuperior = {150,100},
6418   \texttwosuperior = {150, 50},
6419   \textthreesuperior = {150, 50},
6420   \textparagraph = {100, },
6421   \textperiodcentered = {500,300},
6422   \textonequarter = { 50, },
6423   \textonehalf = { 50, },
6424   \textplusminus = {100,100},
6425   \textmultiply = {150,150},
6426   \textdivide = {150,150}
6427 }
6428
6429 </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).

```

6430 <*(blg|ugm)>
6431 \SetProtrusion
6432 <m-t> [ name = OT1-sc,
6433 <bch> [ name = bch-sc,
6434 <cmr> [ name = cmr-sc-OT1,
6435 <pad> [ name = pad-sc,
6436 <pmn> [ name = pmnj-sc,
6437 <ppl> [ name = ppl-sc,
6438 <ptm> [ name = ptm-sc,
6439 <m-t> load = default ]
6440 <bch> load = bch-default ]
6441 <cmr> load = cmr-OT1 ]
6442 <pad> load = pad-default ]
6443 <pmn> load = pmnj-default ]
6444 <ppl> load = ppl-default ]
6445 <ptm> load = ptm-default ]
6446 <m-t|bch|pad|pmn> { encoding = OT1,
6447 <cmr|ppl|ptm> { encoding = {OT1,OT4},
6448 <bch> family = bch,
6449 <cmr> family = cmr,
6450 <pad> family = {pad,padx,padj},

```

```

6451 <pmn>    family = pmnj,
6452 <ppl>    family = {ppl,pplx,pplj},
6453 <ptm>    family = {ptm,ptmx,ptmj},
6454    shape = sc }
6455  {
6456    a = {50,50},
6457 <cmr|pad|ppl|ptm> \ae = {50, },
6458 <bch|pmn>    c = {50, },
6459 <bch|pad|pmn> d = { ,50},
6460 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
6461 <bch|pad|pmn> g = {50, },
6462 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
6463 <bch>        j = {100, },
6464 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
6465 <ptm>        l = { ,80},
6466 <m-t|bch|cmr|pad|pmn|ppl> 013 = { ,50}, % fl
6467 <ptm>        013 = { ,80}, % fl
6468 <bch|pad|pmn> o = {50,50},
6469 <pad|pmn>    \oe = {50, },
6470 <ppl>        p = { 0, 0},
6471 <bch|pad|pmn> q = {50,70},
6472 <ppl>        q = { 0, },
6473 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
6474    t = {50,50},
6475 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
6476 <ptm>        y = {80,80}
6477  }
6478
6479 \SetProtrusion
6480 <m-t> [ name = T1-sc,
6481 <bch> [ name = bch-sc-T1,
6482 <cmr> [ name = cmr-sc-T1,
6483 <pad> [ name = pad-sc-T1,
6484 <pmn> [ name = pmnj-sc-T1,
6485 <ppl> [ name = ppl-sc-T1,
6486 <ptm> [ name = ptm-sc-T1,
6487 <m-t> load = T1-default ]
6488 <bch> load = bch-T1 ]
6489 <cmr> load = cmr-T1 ]
6490 <pad> load = pad-T1 ]
6491 <pmn> load = pmnj-T1 ]
6492 <ppl> load = ppl-T1 ]
6493 <ptm> load = ptm-T1 ]
6494  { encoding = {T1,LY1},
6495 <bch>    family = bch,
6496 <cmr>    family = cmr,
6497 <pad>    family = {pad,padx,padj},
6498 <pmn>    family = pmnj,
6499 <ppl>    family = {ppl,pplx,pplj},
6500 <ptm>    family = {ptm,ptmx,ptmj},
6501    shape = sc }
6502  {
6503    a = {50,50},
6504 <cmr|pad|ppl|ptm> \ae = {50, },
6505 <bch|pmn>    c = {50, },
6506 <bch|pad|pmn> d = { ,50},
6507 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
6508 <bch|pad|pmn> g = {50, },
6509 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
6510 <bch>        j = {100, },
6511 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
6512 <ptm>        l = { ,80},
6513 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl
6514 <ptm>        029 = { ,80}, % fl
6515 <bch|pad|pmn> o = {50,50},

```

```

6516 <bch|pad|pmn> \oe = {50, },
6517 <ppl> p = { 0, 0},
6518 <bch|pad|pmn> q = {50,70},
6519 <ppl> q = { 0, },
6520 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
6521 t = {50,50},
6522 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
6523 <ptm> y = {80,80}
6524 }
6525
6526 <!!(blg|ugm)>
6527 <*m-t|cmr>
6528 \SetProtrusion
6529 <m-t> [ name = T2A-sc,
6530 <cmr> [ name = cmr-sc-T2A,
6531 <m-t> load = T2A-default ]
6532 <cmr> load = cmr-T2A ]
6533 { encoding = T2A,
6534 <cmr> family = cmr,
6535 shape = sc }
6536 {
6537 \cyra = {50,50},
6538 \cyrg = { ,50},
6539 \cyrt = {50,50},
6540 \cyry = { ,50}
6541 }
6542
6543 </m-t|cmr>
6544 <*m-t>
6545 \SetProtrusion
6546 [ name = QX-sc,
6547 load = QX-default ]
6548 { encoding = QX,
6549 shape = sc }
6550 {
6551 a = {50,50},
6552 f = { ,50},
6553 j = {50, },
6554 l = { ,50},
6555 013 = { ,50}, % fl
6556 r = { , 0},
6557 t = {50,50},
6558 y = {50,50}
6559 }
6560
6561 </m-t>
6562 <*cmr|bch>
6563 \SetProtrusion
6564 <bch> [ name = bch-sc-T5,
6565 <bch> load = bch-T5 ]
6566 <cmr> [ name = cmr-sc-T5,
6567 <cmr> load = cmr-T5 ]
6568 { encoding = T5,
6569 <bch> family = bch,
6570 <cmr> family = cmr,
6571 shape = sc }
6572 {
6573 a = {50,50},
6574 <bch> c = {50, },
6575 <bch> d = { ,50},
6576 f = { ,50},
6577 <bch> g = {50, },
6578 <bch> j = {100, },
6579 <cmr> j = {50, },
6580 l = { ,50},

```

```

6581 <bch>    o = {50,50},
6582 <bch>    q = { 0,  },
6583 <cmr>    r = {  , 0},
6584    t = {50,50},
6585    y = {50,50}
6586  }
6587
6588 </cmr|bch>
6589 <*pmn>
6590 \SetProtrusion
6591 [ name    = pmnx-sc,
6592   load    = pmnj-sc ]
6593 { encoding = OT1,
6594   family  = pmnx,
6595   shape   = sc }
6596 {
6597   1 = {230,180}
6598 }
6599
6600 \SetProtrusion
6601 [ name    = pmnx-sc-T1,
6602   load    = pmnj-sc-T1 ]
6603 { encoding = {T1,LY1},
6604   family  = pmnx,
6605   shape   = sc }
6606 {
6607   1 = {230,180}
6608 }
6609

```

#### 15.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's font installation guide suggests `si`.

```

6610 \SetProtrusion
6611 [ name    = pmnj-scit,
6612   load    = pmnj-it  ]
6613 { encoding = OT1,
6614   family  = pmnj,
6615   shape   = {scit,si} }
6616 {
6617   a = {50,  },
6618   \ae = {  , -50},
6619   b = {20,-50},
6620   c = {50,-50},
6621   d = {20, 0},
6622   e = {20,-50},
6623   f = {10, 0},
6624   012 = {10,-50}, % fi
6625   013 = {10,-50}, % fl
6626   014 = {10,-50}, % ffi
6627   015 = {10,-50}, % ffl
6628   g = {50,-50},
6629   i = {20,-50},
6630   j = {20, 0},
6631   k = {20,  },
6632   l = {20,50},
6633   m = {  , -30},
6634   n = {  , -30},
6635   o = {50,  },
6636   \oe = {50,-50},
6637   p = {20,-50},
6638   q = {50,  },
6639   r = {20, 0},

```

```

6640     s = {20,-30},
6641     t = {70, },
6642     u = {50,-50},
6643     v = {100, },
6644     w = {100, },
6645     y = {50, },
6646     z = { , -50}
6647 }
6648
6649 \SetProtrusion
6650 [ name = pmnj-scit-T1,
6651   load = pmnj-it-T1 ]
6652 { encoding = {T1,LY1},
6653   family = pmnj,
6654   shape = {scit,si} }
6655 {
6656   a = {50, },
6657   \ae = { , -50},
6658   b = {20,-50},
6659   c = {50,-50},
6660   d = {20, 0},
6661   e = {20,-50},
6662   f = {10, 0},
6663   028 = {10,-50}, % fi
6664   029 = {10,-50}, % fl
6665   030 = {10,-50}, % ffi
6666   031 = {10,-50}, % ffl
6667   g = {50,-50},
6668   i = {20,-50},
6669   188 = {20, 0}, % ij
6670   j = {20, 0},
6671   k = {20, },
6672   l = {20,50},
6673   m = { , -30},
6674   n = { , -30},
6675   o = {50, },
6676   \oe = {50,-50},
6677   p = {20,-50},
6678   q = {50, },
6679   r = {20, 0},
6680   s = {20,-30},
6681   t = {70, },
6682   u = {50,-50},
6683   v = {100, },
6684   w = {100, },
6685   y = {50, },
6686   z = { , -50}
6687 }
6688
6689 \SetProtrusion
6690 [ name = pmnx-scit,
6691   load = pmnj-scit ]
6692 { encoding = OT1,
6693   family = pmnx,
6694   shape = {scit,si} }
6695 {
6696   l = {100,150}
6697 }
6698
6699 \SetProtrusion
6700 [ name = pmnx-scit-T1,
6701   load = pmnj-scit-T1 ]
6702 { encoding = {T1,LY1},
6703   family = pmnx,
6704   shape = {scit,si} }

```

```

6705 {
6706   1 = {100,150}
6707 }
6708
6709 /pmn)

```

### 15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

6710 \SetProtrusion
6711 <m-t> [ name = textcomp ]
6712 <bch> [ name = bch-textcomp ]
6713 <blg> [ name = blg-textcomp ]
6714 <cmr> [ name = cmr-textcomp ]
6715 <pad> [ name = pad-textcomp ]
6716 <pmn> [ name = pmn-textcomp ]
6717 <ppl> [ name = ppl-textcomp ]
6718 <ptm> [ name = ptm-textcomp ]
6719 <ugm> [ name = ugm-textcomp ]
6720 <m-t> { encoding = TS1 }
6721 <!m-t> { encoding = TS1,
6722 <bch> family = bch }
6723 <blg> family = blg }
6724 <cmr> family = cmr }
6725 <pad> family = {pad,padx,padj} }
6726 <pmn> family = {pmnx,pmnj} }
6727 <ppl> family = {ppl,pplx,pplj} }
6728 <ptm> family = {ptm,ptmx,ptmj} }
6729 <ugm> family = ugm }
6730 {
6731 <blg> \textquotestraightbase = {400,500},
6732 <cmr> \textquotestraightbase = {300,300},
6733 <pad|pmn> \textquotestraightbase = {400,400},
6734 <blg> \textquotestraightdblbase = {300,400},
6735 <cmr|pmn> \textquotestraightdblbase = {300,300},
6736 <pad> \textquotestraightdblbase = {400,400},
6737 <bch|cmr|pad|pmn|ugm> \texttwelveudash = {200,200},
6738 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
6739 <ugm> \textthreequartersemdash = {200,200},
6740 <blg> \textquotesingle = {500,600},
6741 <cmr|pmn> \textquotesingle = {300,400},
6742 <pad> \textquotesingle = {400,500},
6743 <ptm> \textquotesingle = {500,500},
6744 <ugm> \textquotesingle = {300,500},
6745 <bch|cmr|pmn> \textasteriskcentered = {200,300},
6746 <blg> \textasteriskcentered = {150,200},
6747 <pad> \textasteriskcentered = {300,300},
6748 <ugm> \textasteriskcentered = {100,200},
6749 <pmn> \textfractionsolidus = {-200,-200},
6750 <cmr> \textoneoldstyle = {100,100},
6751 <pmn> \textoneoldstyle = { , 50},
6752 <cmr> \textthreeoldstyle = { , 50},
6753 <pad|pmn> \textthreeoldstyle = { 50, },
6754 <cmr> \textfouroldstyle = { 50, 50},
6755 <pad|pmn> \textfouroldstyle = { 50, },
6756 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
6757 <cmr> \textlangle = {400, },
6758 <cmr> \textrightangle = { ,400},
6759 <m-t|bch|pmn|ptm> \textminus = {200,200},
6760 <cmr|pad|ppl> \textminus = {300,300},
6761 <blg|ugm> \textminus = {250,300},
6762 <bch|pad|pmn> \textlbrackdbl = {100, },
6763 <blg> \textlbrackdbl = {200, },

```

```

6764 <bch|pad|pmn> \textrbrackdbl = { ,100},
6765 <blg> \textrbrackdbl = { ,200},
6766 <pmn> \textasciigrave = {200,500},
6767 <bch|blg|cmr|pad|pmn> \texttildebelow = {200,250},
6768 <pmn> \textasciibreve = {300,400},
6769 <pmn> \textasciicaron = {300,400},
6770 <pmn> \textacutedbl = {200,300},
6771 <pmn> \textgravedbl = {150,300},
6772 <bch|pmn|ugm> \textdagger = { 80, 80},
6773 <blg> \textdagger = {200,200},
6774 <cmr|pad> \textdagger = {100,100},
6775 <ptm> \textdagger = {150,150},
6776 <blg> \textdaggerdbl = {150,150},
6777 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
6778 <ptm> \textdaggerdbl = {100,100},
6779 <bch> \textbardbl = {100,100},
6780 <blg|ugm> \textbardbl = {150,150},
6781 <bch> \textbullet = {200,200},
6782 <blg> \textbullet = {400,500},
6783 <cmr|pad|pmn> \textbullet = { ,100},
6784 <ptm> \textbullet = {150,150},
6785 <ugm> \textbullet = { 50,100},
6786 <bch|cmr|pmn> \textcelsius = { 50, },
6787 <pad> \textcelsius = { 80, },
6788 <bch> \textflorin = { 50, 50},
6789 <blg> \textflorin = {100,100},
6790 <pad|ugm> \textflorin = { ,100},
6791 <pmn> \textflorin = { 50,100},
6792 <ptm> \textflorin = { 50, 70},
6793 <cmr> \textcolonmonetary = { , 50},
6794 <pad|pmn> \textcolonmonetary = { 50, },
6795 <pmn> \textinterrobang = { ,100},
6796 <pmn> \textinterrobangdown = {100, },
6797 <m-t|pad|ptm> \texttrademark = {100,100},
6798 <bch> \texttrademark = {150,150},
6799 <blg|cmr|ppl> \texttrademark = {200,200},
6800 <pmn> \texttrademark = { 50, 50},
6801 <ugm> \texttrademark = {100,150},
6802 <bch|ugm> \textcent = { 50, },
6803 <ptm> \textcent = {100,100},
6804 <bch> \textsterling = { 50, },
6805 <ugm> \textsterling = { , 50},
6806 <bch> \textbrokenbar = {200,200},
6807 <blg> \textbrokenbar = {250,250},
6808 <ugm> \textbrokenbar = {200,300},
6809 <pmn> \textasciidieresis = {300,400},
6810 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
6811 <pmn> \textcopyright = {100,150},
6812 <ppl> \textcopyright = {200,200},
6813 <bch|cmr|ugm> \textordfeminine = {100,200},
6814 <pad|pmn> \textordfeminine = {200,200},
6815 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
6816 <blg> \textlnot = {200,100},
6817 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
6818 <pmn> \textregistered = { 50,150},
6819 <ppl> \textregistered = {200,200},
6820 <pmn> \textasciimacron = {150,200},
6821 <m-t|ppl|ptm> \textdegree = {300,300},
6822 <bch> \textdegree = {150,200},
6823 <blg|ugm> \textdegree = {200,200},
6824 <cmr|pad> \textdegree = {400,400},
6825 <pmn> \textdegree = {150,400},
6826 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
6827 <blg> \textpm = {100,100},
6828 <ptm> \textpm = { 50, 80},

```

```

6829 <bch|blg|ugm> \texttwosuperior = {100,200},
6830 <cmr> \texttwosuperior = { 50,100},
6831 <pad|pmn> \texttwosuperior = {200,200},
6832 <ptm> \texttwosuperior = { 50, 50},
6833 <bch|blg|ugm> \textthreesuperior = {100,200},
6834 <cmr> \textthreesuperior = { 50,100},
6835 <pad|pmn> \textthreesuperior = {200,200},
6836 <ptm> \textthreesuperior = { 50, 50},
6837 <pmn> \textasciiaacute = {300,400},
6838 <bch|ugm> \textmu = { ,100},
6839 <bch|pad|pmn> \textparagraph = { ,100},
6840 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
6841 <blg> \textperiodcentered = {400,500},
6842 <ptm> \textperiodcentered = {300,300},
6843 <ugm> \textperiodcentered = {200,500},
6844 <bch|blg|ugm> \textonesuperior = {200,300},
6845 <cmr|pad|pmn> \textonesuperior = {200,200},
6846 <ptm> \textonesuperior = {100,100},
6847 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
6848 <blg|cmr> \textordmasculine = {100,200},
6849 <bch|cmr|pmn> \texteuro = {100, },
6850 <pad> \texteuro = { 50,100},
6851 <bch> \texttimes = {200,200},
6852 <blg|ptm> \texttimes = {100,100},
6853 <cmr> \texttimes = {150,250},
6854 <pad> \texttimes = {100,150},
6855 <pmn> \texttimes = { 70,100},
6856 <ugm> \texttimes = {200,300},
6857 <bch|pad|pmn> \textdiv = {150,200}
6858 <blg> \textdiv = {100,100}
6859 <cmr> \textdiv = {150,250}
6860 <ptm> \textdiv = { 50,100},
6861 <ugm> \textdiv = {200,300},
6862 <ptm> \textperthousand = { ,50}
6863 <ugm> \textsection = { ,100},
6864 <ugm> \textonehalf = { 50,100},
6865 <ugm> \textonequarter = { 50,100},
6866 <ugm> \textthreequarters = { 50,100},
6867 <ugm> \textsurd = { ,100}

```

Remaining slots in the source file.

```

6868 }
6869
6870 <*cmr|pad|pmn|ugm>
6871 \SetProtrusion
6872 <cmr> [ name = cmr-textcomp-it ]
6873 <pad> [ name = pad-textcomp-it ]
6874 <pmn> [ name = pmn-textcomp-it ]
6875 <ugm> [ name = ugm-textcomp-it ]
6876 { encoding = TS1,
6877 <cmr> family = cmr,
6878 <pad> family = {pad,padx,padj},
6879 <pmn> family = {pmnx,pmnj},
6880 <ugm> family = ugm,
6881 <!ugm> shape = {it,s1} }
6882 <ugm> shape = it }
6883 {
6884 <cmr> \textquotestraightbase = {300,600},
6885 <pad|pmn> \textquotestraightbase = {400,400},
6886 <cmr> \textquotestraightdblbase = {300,600},
6887 <pad> \textquotestraightdblbase = {300,400},
6888 <pmn> \textquotestraightdblbase = {300,300},
6889 \texttwelveudash = {200,200},
6890 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
6891 <ugm> \textthreequartersemdash = {200,200},

```

```

6892 <cmr> \textquotesingle = {600,300},
6893 <pad> \textquotesingle = {800,100},
6894 <pmn> \textquotesingle = {300,200},
6895 <ugm> \textquotesingle = {500,500},
6896 <cmr> \textasteriskcentered = {300,200},
6897 <pad> \textasteriskcentered = {500,100},
6898 <pmn> \textasteriskcentered = {200,300},
6899 <ugm> \textasteriskcentered = {300,150},
6900 <pmn> \textfractionsolidus = {-200,-200},
6901 <cmr> \textoneoldstyle = {100, 50},
6902 <pad> \textoneoldstyle = {100, },
6903 <pmn> \textoneoldstyle = { 50, },
6904 <pad> \texttwooldstyle = { 50, },
6905 <pmn> \texttwooldstyle = {-50, },
6906 <cmr> \textthreeoldstyle = {100, 50},
6907 <pmn> \textthreeoldstyle = {-100, },
6908 <cmr> \textfouroldstyle = { 50, 50},
6909 <pad> \textfouroldstyle = { 50,100},
6910 <cmr> \textsevenoldstyle = { 50, 80},
6911 <pad> \textsevenoldstyle = { 50, },
6912 <pmn> \textsevenoldstyle = { 20, },
6913 <cmr> \textlangle = {400, },
6914 <cmr> \textrangle = { ,400},
6915 <cmr|pad> \textminus = {300,300},
6916 <pmn> \textminus = {200,200},
6917 <ugm> \textminus = {250,300},
6918 <pad|pmn> \textlbrackdbl = {100, },
6919 <pad|pmn> \textrbrackdbl = { ,100},
6920 <pmn> \textasciigrave = {300,300},
6921 <cmr|pad|pmn> \texttildelow = {200,250},
6922 <pmn> \textasciibreve = {300,300},
6923 <pmn> \textasciicaron = {300,300},
6924 <pmn> \textacutedbl = {200,300},
6925 <pmn> \textgravedbl = {150,300},
6926 <cmr> \textdagger = {100,100},
6927 <pad> \textdagger = {200,100},
6928 <pmn> \textdagger = { 80, 50},
6929 <ugm> \textdagger = { 80, 80},
6930 <cmr|pad> \textdaggerdbl = { 80, 80},
6931 <pmn> \textdaggerdbl = { 80, 50},
6932 <ugm> \textbardbl = {150,150},
6933 <cmr> \textbullet = {200,100},
6934 <pad> \textbullet = {300, },
6935 <pmn> \textbullet = { 30, 70},
6936 <ugm> \textbullet = { 50,100},
6937 <cmr> \textcelsius = {100, },
6938 <pad> \textcelsius = {200, },
6939 <pmn> \textcelsius = { 50,-50},
6940 <pad> \textflorin = {100, },
6941 <pmn> \textflorin = { 50,100},
6942 <ugm> \textflorin = { ,100},
6943 <cmr> \textcolonmonetary = {150, },
6944 <pad> \textcolonmonetary = {100, },
6945 <pmn> \textcolonmonetary = { 50,-50},
6946 <cmr|pad> \texttrademark = {200, },
6947 <pmn> \texttrademark = { 50,100},
6948 <ugm> \texttrademark = {150, 50},
6949 <ugm> \textcent = { 50, },
6950 <ugm> \textsterling = { , 50},
6951 <ugm> \textbrokenbar = {200,300},
6952 <pmn> \textasciidieresis = {300,200},
6953 <cmr> \textcopyright = {100, },
6954 <pad> \textcopyright = {200,100},
6955 <pmn> \textcopyright = {100,150},
6956 <ugm> \textcopyright = {300, },

```

```

6957 <cmr> \textordfeminine = {100,100},
6958 <pmn> \textordfeminine = {200,200},
6959 <ugm> \textordfeminine = {100,200},
6960 <cmr|pad> \textlnot = {300, },
6961 <pmn|ugm> \textlnot = {200, },
6962 <cmr> \textregistered = {100, },
6963 <pad> \textregistered = {200,100},
6964 <pmn> \textregistered = { 50,150},
6965 <ugm> \textregistered = {300, },
6966 <pmn> \textasciimacron = {150,200},
6967 <cmr|pad> \textdegree = {500,100},
6968 <pmn> \textdegree = {150,150},
6969 <ugm> \textdegree = {300,200},
6970 <cmr> \textpm = {150,100},
6971 <pad> \textpm = {200,150},
6972 <pmn|ugm> \textpm = {150,200},
6973 <cmr> \textonesuperior = {400, },
6974 <pad> \textonesuperior = {300,100},
6975 <pmn> \textonesuperior = {200,100},
6976 <ugm> \textonesuperior = {300,300},
6977 <cmr> \texttwosuperior = {400, },
6978 <pad> \texttwosuperior = {300, },
6979 <pmn> \texttwosuperior = {200,100},
6980 <ugm> \texttwosuperior = {300,200},
6981 <cmr> \textthreesuperior = {400, },
6982 <pad> \textthreesuperior = {300, },
6983 <pmn> \textthreesuperior = {200,100},
6984 <ugm> \textthreesuperior = {300,200},
6985 <ugm> \textmu = { ,100},
6986 <pmn> \textasciiacute = {300,200},
6987 <cmr> \textparagraph = {200, },
6988 <pmn> \textparagraph = { ,100},
6989 <cmr> \textperiodcentered = {500,500},
6990 <pad|pmn|ugm> \textperiodcentered = {300,400},
6991 <cmr> \textordmasculine = {100,100},
6992 <pmn> \textordmasculine = {200,200},
6993 <ugm> \textordmasculine = {300,200},
6994 <cmr> \texteuro = {200, },
6995 <pad> \texteuro = {100, },
6996 <pmn> \texteuro = {100,-50},
6997 <cmr> \texttimes = {200,200},
6998 <pad> \texttimes = {200,100},
6999 <pmn> \texttimes = { 70,100},
7000 <ugm> \texttimes = {200,300},
7001 <cmr|pad> \textdiv = {200,200}
7002 <pmn> \textdiv = {150,200}
7003 <ugm> \textdiv = {200,300},
7004 <ugm> \textsection = { ,200},
7005 <ugm> \textonehalf = { 50,100},
7006 <ugm> \textonequarter = { 50,100},
7007 <ugm> \textthreequarters = { 50,100},
7008 <ugm> \textsurd = { ,100}
7009 }
7010
7011 </cmr|pad|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}
```

```
7012 <*cmr>
7013 \SetProtrusion
7014 [ name = cmr-math-letters ]
7015 { encoding = OML,
7016   family = cmm,
7017   series = {m,b},
7018   shape = it }
7019 {
7020   A = {100, 50}, % \mathnormal
7021   B = { 50,  },
7022   C = { 50,  },
7023   D = { 50, 50},
7024   E = { 50,  },
7025   F = {100, 50},
7026   G = { 50, 50},
7027   H = { 50, 50},
7028   I = { 50, 50},
7029   J = {150, 50},
7030   K = { 50,100},
7031   L = { 50, 50},
7032   M = { 50,  },
7033   N = { 50,  },
7034   O = { 50,  },
7035   P = { 50,  },
7036   Q = { 50, 50},
7037   R = { 50,  },
7038   S = { 50,  },
7039   T = { 50,100},
7040   U = { 50, 50},
7041   V = {100,100},
7042   W = { 50,100},
7043   X = { 50,100},
7044   Y = {100,100},
7045   f = {100,100},
7046   h = {  ,100},
7047   i = {  , 50},
7048   j = {  , 50},
7049   k = {  , 50},
7050   r = {  , 50},
7051   v = {  , 50},
7052   w = {  , 50},
7053   x = {  , 50},
7054   "0B = { 50,100}, % \alpha
7055   "0C = { 50, 50}, % \beta
7056   "0D = {200,150}, % \gamma
7057   "0E = { 50, 50}, % \delta
7058   "0F = { 50, 50}, % \epsilon
7059   "10 = { 50,150}, % \zeta
7060   "12 = { 50,  }, % \theta
7061   "13 = {  ,100}, % \iota
7062   "14 = {  ,100}, % \kappa
7063   "15 = {100, 50}, % \lambda
7064   "16 = {  , 50}, % \mu
7065   "17 = {  , 50}, % \nu
```

```

7066 "18 = { , 50}, % \xi
7067 "19 = { 50,100}, % \pi
7068 "1A = { 50, 50}, % \rho
7069 "1B = { ,150}, % \sigma
7070 "1C = { 50,150}, % \tau
7071 "1D = { 50, 50}, % \upsilon
7072 "1F = { 50,100}, % \chi
7073 "20 = { 50, 50}, % \psi
7074 "21 = { , 50}, % \omega
7075 "22 = { , 50}, % \varepsilon
7076 "23 = { , 50}, % \vartheta
7077 "24 = { , 50}, % \varpi
7078 "25 = {100, }, % \varrho
7079 "26 = {100,100}, % \varsigma
7080 "27 = { 50, 50}, % \varphi
7081 "28 = {100,100}, % \leftharpoonup
7082 "29 = {100,100}, % \leftharpoondown
7083 "2A = {100,100}, % \rightharpoonup
7084 "2B = {100,100}, % \rightharpoondown
7085 "2C = {300,200}, % \lhook
7086 "2D = {200,300}, % \rhook
7087 "2E = { ,100}, % \triangleright
7088 "2F = {100, }, % \triangleleft
7089 "3A = { ,500}, % ., \ldotp
7090 "3B = { ,500}, % ,
7091 "3C = {200,100}, % <
7092 "3D = {300,400}, % /
7093 "3E = {100,200}, % >
7094 "3F = {200,200}, % \star
7095 "5B = { ,100}, % \flat
7096 "5E = {200,200}, % \smile
7097 "5F = {200,200}, % \frown
7098 "7C = {100, }, % \jmath
7099 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

7100 }
7101

```

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}

```

```

7102 \SetProtrusion
7103 [ name = cmr-math-symbols ]
7104 { encoding = OMS,
7105   family = cmsy,
7106   series = {m,b},
7107   shape = n }
7108 {
7109   A = {150, 50}, % \mathcal
7110   C = { ,100},
7111   D = { , 50},
7112   F = { 50,150},
7113   I = { ,100},
7114   J = {100,150},
7115   K = { ,100},
7116   L = {100, },
7117   M = { 50, 50},
7118   N = { 50,100},
7119   P = { , 50},
7120   Q = { 50, },
7121   R = { , 50},
7122   T = { 50,150},
7123   V = { 50, 50},

```

```

7124     W = { , 50},
7125     X = {100,100},
7126     Y = {100, },
7127     Z = {100,150},
7128     "00 = {300,300}, % -
7129     "01 = { ,700}, % \cdot, \cdotp
7130     "02 = {150,250}, % \times
7131     "03 = {150,250}, % *, \ast
7132     "04 = {200,300}, % \div
7133     "05 = {150,250}, % \diamond
7134     "06 = {200,200}, % \pm
7135     "07 = {200,200}, % \mp
7136     "08 = {100,100}, % \oplus
7137     "09 = {100,100}, % \ominus
7138     "0A = {100,100}, % \otimes
7139     "0B = {100,100}, % \oslash
7140     "0C = {100,100}, % \odot
7141     "0D = {100,100}, % \bigcirc
7142     "0E = {100,100}, % \circ
7143     "0F = {100,100}, % \bullet
7144     "10 = {100,100}, % \asymp
7145     "11 = {100,100}, % \equiv
7146     "12 = {200,100}, % \subseteq
7147     "13 = {100,200}, % \supseteq
7148     "14 = {200,100}, % \leq
7149     "15 = {100,200}, % \geq
7150     "16 = {200,100}, % \preceq
7151     "17 = {100,200}, % \succeq
7152     "18 = {200,200}, % \sim
7153     "19 = {150,150}, % \approx
7154     "1A = {200,100}, % \subset
7155     "1B = {100,200}, % \supset
7156     "1C = {200,100}, % \ll
7157     "1D = {100,200}, % \gg
7158     "1E = {300,100}, % \prec
7159     "1F = {100,300}, % \succ
7160     "20 = {100,200}, % \leftarrow
7161     "21 = {200,100}, % \rightarrow
7162     "22 = {100,100}, % \uparrow
7163     "23 = {100,100}, % \downarrow
7164     "24 = {100,100}, % \leftrightarrows
7165     "25 = {100,100}, % \nearrow
7166     "26 = {100,100}, % \searrow
7167     "27 = {100,100}, % \simeq
7168     "28 = {100,100}, % \Leftarrow
7169     "29 = {100,100}, % \Rightarrow
7170     "2A = {100,100}, % \Uparrow
7171     "2B = {100,100}, % \Downarrow
7172     "2C = {100,100}, % \Leftrightarrow
7173     "2D = {100,100}, % \nrightarrow
7174     "2E = {100,100}, % \swarrow
7175     "2F = { ,100}, % \propto
7176     "30 = { ,400}, % \prime
7177     "31 = {100,100}, % \infty
7178     "32 = {150,100}, % \in
7179     "33 = {100,150}, % \ni
7180     "34 = {100,100}, % \triangle, \bigtriangleup
7181     "35 = {100,100}, % \bigtriangledown
7182     "38 = { ,100}, % \forall
7183     "39 = {100, }, % \exists
7184     "3A = {200, }, % \neg
7185     "3E = {200,200}, % \top
7186     "3F = {200,200}, % \bot, \perp
7187     "5E = {100,200}, % \wedge
7188     "5F = {100,200}, % \vee

```

```

7189 "60 = { ,300}, % \vdash
7190 "61 = {300, }, % \dashv
7191 "62 = {100,100}, % \lfloor
7192 "63 = {100,100}, % \rfloor
7193 "64 = {100,100}, % \lceil
7194 "65 = {100,100}, % \rceil
7195 "66 = {150, }, % \lbrace
7196 "67 = { ,150}, % \rbrace
7197 "68 = {400, }, % \langle
7198 "69 = { ,400}, % \rangle
7199 "6C = {100,100}, % \updownarrow
7200 "6D = {100,100}, % \Updownarrow
7201 "6E = {100,300}, % \, \backslash, \setminus
7202 "72 = {100,100}, % \nabla
7203 "79 = {200,200}, % \dagger
7204 "7A = {100,100}, % \ddagger
7205 "7B = {100, }, % \mathparagraph
7206 "7C = {100,100}, % \clubsuit
7207 "7D = {100,100}, % \diamondsuit
7208 "7E = {100,100}, % \heartsuit
7209 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

7210 }
7211

```

We don't bother about 'largsymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largsymbols}{OMX}{cmex}{m}{n}
```

```

7212 </cmr>
7213 </cfg-t>

```

### 15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
7214 <*cfg-u>
```

Symbol font 'a'.

```

7215 <*msa>
7216 \SetProtrusion
7217 [ name = AMS-a ]
7218 { encoding = U,
7219 family = msa }
7220 {
7221 "05 = {150,250}, % \centerdot
7222 "06 = {100,100}, % \lozenge
7223 "07 = { 50, 50}, % \blacklozenge
7224 "08 = { 50, 50}, % \circlearrowright
7225 "09 = { 50, 50}, % \circlearrowleft
7226 "0A = {100,100}, % \rightleftharpoons
7227 "0B = {100,100}, % \leftrightharpoons
7228 "0D = {-50,200}, % \Vdash
7229 "0E = {-50,200}, % \Vvdash
7230 "0F = {-70,150}, % \vDash
7231 "10 = {100,150}, % \twoheadrightarrow
7232 "11 = {100,150}, % \twoheadleftarrow
7233 "12 = { 50,100}, % \leftleftarrows
7234 "13 = { 50, 80}, % \rightrightarrows
7235 "14 = {120,120}, % \upuparrows
7236 "15 = {120,120}, % \downdownarrows
7237 "16 = {200,200}, % \upharpoonright
7238 "17 = {200,200}, % \downharpoonright

```

```

7239 "18 = {200,200}, % \upharpoonleft
7240 "19 = {200,200}, % \downharpoonleft
7241 "1A = { 80,100}, % \rightarrowtail
7242 "1B = { 80,100}, % \leftarrowtail
7243 "1C = { 50, 50}, % \leftrightarrows
7244 "1D = { 50, 50}, % \rightleftarrows
7245 "1E = {250,  }, % \Lsh
7246 "1F = {  ,250}, % \Rsh
7247 "20 = {100,100}, % \rightsquigarrow
7248 "21 = {100,100}, % \leftrightsquigarrow
7249 "22 = {100, 50}, % \looparrowleft
7250 "23 = { 50,100}, % \looparrowright
7251 "24 = { 50, 80}, % \circeq
7252 "25 = {  ,100}, % \succsim
7253 "26 = {  ,100}, % \gtrsim
7254 "27 = {  ,100}, % \gtrapprox
7255 "28 = {150, 50}, % \multimap
7256 "2B = {100,150}, % \doteqdot
7257 "2C = {100,150}, % \triangleq
7258 "2D = {100, 50}, % \precsim
7259 "2E = {100, 50}, % \lessim
7260 "2F = { 50, 50}, % \lessapprox
7261 "30 = {100, 50}, % \eqslantless
7262 "31 = { 50, 50}, % \eqslantgtr
7263 "32 = {100, 50}, % \curlyeqprec
7264 "33 = { 50,100}, % \curlyeqsucc
7265 "34 = {100, 50}, % \preccurlyeq
7266 "36 = { 50,  }, % \leqslant
7267 "38 = {  , 50}, % \backprime
7268 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
7269 "3C = { 50,100}, % \succcurlyeq
7270 "3E = {  , 50}, % \geqslant
7271 "40 = {  , 50}, % \sqsubset
7272 "41 = { 50,  }, % \sqsupset
7273 "42 = {  ,150}, % \vartriangleright, \rhd
7274 "43 = {150,  }, % \vartriangleleft, \lhd
7275 "44 = {  ,100}, % \trianglerighteq, \unrhd
7276 "45 = {100,  }, % \trianglelefteq, \unlhd
7277 "46 = {100,100}, % \bigstar
7278 "48 = { 50, 50}, % \blacktriangledown
7279 "49 = {  ,100}, % \blacktriangleright
7280 "4A = {100,  }, % \blacktriangleleft
7281 "4B = {  ,150}, % \dashrightarrow (the arrow)
7282 "4C = {150,  }, % \dashleftarrow
7283 "4D = { 50, 50}, % \vartriangle
7284 "4E = { 50, 50}, % \blacktriangle
7285 "4F = { 50, 50}, % \triangledown
7286 "50 = { 50, 50}, % \eqcirc
7287 "56 = {  ,150}, % \Rrightarrow
7288 "57 = {150,  }, % \Lleftarrow
7289 "58 = {100,300}, % \checkmark
7290 "5C = { 50, 50}, % \angle
7291 "5D = { 50, 50}, % \measuredangle
7292 "5E = { 50, 50}, % \sphericalangle
7293 "5F = {  , 50}, % \varpropto
7294 "60 = {100,100}, % \smallsmile
7295 "61 = {100,100}, % \smallfrown
7296 "62 = { 50,  }, % \Subset
7297 "63 = {  , 50}, % \Supset
7298 "66 = {150,150}, % \curlywedge
7299 "67 = {150,150}, % \curlyvee
7300 "68 = { 50,150}, % \leftthreetimes
7301 "69 = {100, 50}, % \rightthreetimes
7302 "6C = { 50, 50}, % \bumpeq
7303 "6D = { 50, 50}, % \Bumpeq

```

```

7304 "6E = {100, }, % \l11
7305 "6F = { ,100}, % \ggg
7306 "70 = { 50,100}, % \ulcorner
7307 "71 = {100, 50}, % \urcorner
7308 "75 = {150,200}, % \dotplus
7309 "76 = { 50,100}, % \backsim
7310 "78 = { 50,100}, % \llcorner
7311 "79 = {100, 50}, % \lrcorner
7312 "7C = {100,100}, % \intercal
7313 "7D = { 50, 50}, % \circledcirc
7314 "7E = { 50, 50}, % \circledast
7315 "7F = { 50, 50} % \circleddash

```

Remaining slots in the source file.

```

7316 }
7317
7318 </msa>

```

Symbol font 'b'.

```

7319 <*msb>
7320 \SetProtrusion
7321 [ name = AMS-b ]
7322 { encoding = U,
7323   family = msb }
7324 {
7325   A = { 50, 50}, % \mathbb
7326   C = { 50, 50},
7327   G = { , 50},
7328   L = { , 50},
7329   P = { , 50},
7330   R = { , 50},
7331   T = { , 50},
7332   V = { 50, 50},
7333   X = { 50, 50},
7334   Y = { 50, 50},
7335 "00 = { 50, 50}, % \lvertneqq
7336 "01 = { 50, 50}, % \gvertneqq
7337 "02 = { 50, 50}, % \nleq
7338 "03 = { 50, 50}, % \ngeq
7339 "04 = {100, 50}, % \nless
7340 "05 = { 50,150}, % \ngtr
7341 "06 = {100, 50}, % \nprec
7342 "07 = { 50,150}, % \nsucc
7343 "08 = { 50, 50}, % \lneqq
7344 "09 = { 50, 50}, % \gneqq
7345 "0A = {100,100}, % \lneqslant
7346 "0B = {100,100}, % \ngeqslant
7347 "0C = {100, 50}, % \lneq
7348 "0D = { 50,100}, % \gneq
7349 "0E = {100, 50}, % \npreceq
7350 "0F = { 50,100}, % \nsucceq
7351 "10 = { 50, }, % \precnsim
7352 "11 = { 50, 50}, % \succnsim
7353 "12 = { 50, 50}, % \lnsim
7354 "13 = { 50, 50}, % \gnsim
7355 "14 = { 50, 50}, % \nleqq
7356 "15 = { 50, 50}, % \ngeqq
7357 "16 = { 50, 50}, % \precneqq
7358 "17 = { 50, 50}, % \succneqq
7359 "18 = { 50, 50}, % \precnapprox
7360 "19 = { 50, 50}, % \succnapprox
7361 "1A = { 50, 50}, % \lnapprox
7362 "1B = { 50, 50}, % \gnapprox
7363 "1C = {150,200}, % \nsim
7364 "1D = { 50, 50}, % \ncong

```

```

7365 "1E = {100,150}, % \diagup
7366 "1F = {100,150}, % \diagdown
7367 "20 = {100, 50}, % \varsubsetneq
7368 "21 = { 50,100}, % \varsupsetneq
7369 "22 = {100, 50}, % \subsetteqq
7370 "23 = { 50,100}, % \supsetteqq
7371 "24 = {100, 50}, % \subseteqq
7372 "25 = { 50,100}, % \supseteqq
7373 "26 = {100, 50}, % \varsubsetneqq
7374 "27 = { 50,100}, % \varsupsetneqq
7375 "28 = {100, 50}, % \subseteq
7376 "29 = { 50,100}, % \supseteq
7377 "2A = {100, 50}, % \subset
7378 "2B = { 50,100}, % \supset
7379 "2C = { 50,100}, % \nparallel
7380 "2D = {100,150}, % \nmid
7381 "2E = {150,150}, % \shortmid
7382 "2F = {100,100}, % \shortparallel
7383 "30 = { ,150}, % \nvDash
7384 "31 = { ,150}, % \nVDash
7385 "32 = { ,100}, % \nvDash
7386 "33 = { ,100}, % \nVDash
7387 "34 = { ,100}, % \ntrianglerighteq
7388 "35 = {100, }, % \trianglelefteq
7389 "36 = {100, }, % \triangleleft
7390 "37 = { ,100}, % \triangleright
7391 "38 = {100,200}, % \leftarrow
7392 "39 = {100,200}, % \rightarrow
7393 "3A = {100,100}, % \Leftarrow
7394 "3B = { 50,100}, % \Rightarrow
7395 "3C = {100,100}, % \Leftrightarrow
7396 "3D = {100,200}, % \leftrightarrows
7397 "3E = { 50, 50}, % \divideontimes
7398 "3F = { 50, 50}, % \varnothing
7399 "60 = {200, }, % \Finv
7400 "61 = { , 50}, % \Game
7401 "68 = {100,100}, % \eqsim
7402 "69 = { 50, }, % \beth
7403 "6A = { 50, }, % \gimel
7404 "6B = {150, }, % \daleth
7405 "6C = {200, }, % \lessdot
7406 "6D = { ,200}, % \gtrdot
7407 "6E = {100,200}, % \ltimes
7408 "6F = {150,100}, % \rtimes
7409 "70 = { 50,100}, % \shortmid
7410 "71 = { 50, 50}, % \shortparallel
7411 "72 = {200,300}, % \smallsetminus
7412 "73 = {100,200}, % \thicksim
7413 "74 = { 50,100}, % \thickapprox
7414 "75 = { 50, 50}, % \approx
7415 "76 = { 50,100}, % \succapprox
7416 "77 = { 50, 50}, % \precapprox
7417 "78 = {100,100}, % \curvearrowleft
7418 "79 = { 50,150}, % \curvearrowright
7419 "7A = { 50,200}, % \digamma
7420 "7B = {100, 50}, % \varkappa
7421 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

7422 }
7423
7424 (/msb)

```

**15.8.8 Euler**

Euler Roman font (package euler).

```

7425 (*eur)
7426 \SetProtrusion
7427 [ name = euler ]
7428 { encoding = U,
7429   family = eur }
7430 {
7431   "01 = {100,100},
7432   "03 = {100,150},
7433   "06 = { ,100},
7434   "07 = {100,150},
7435   "08 = {100,100},
7436   "0A = {100,100},
7437   "0B = { , 50},
7438   "0C = { ,100},
7439   "0D = {100,100},
7440   "0E = { ,100},
7441   "0F = {100,100},
7442   "10 = {100,100},
7443   "13 = { ,100},
7444   "14 = { ,100},
7445   "15 = { , 50},
7446   "16 = { , 50},
7447   "17 = { 50,100},
7448   "18 = { 50,100},
7449   "1A = { , 50},
7450   "1B = { , 50},
7451   "1C = { 50,100},
7452   "1D = { 50,100},
7453   "1E = { 50,100},
7454   "1F = { 50,100},
7455   "20 = { , 50},
7456   "21 = { , 50},
7457   "22 = { 50,100},
7458   "24 = { , 50},
7459   "27 = { 50,100},
7460   1 = {100,100},
7461   7 = { 50,100},
7462   "3A = {300,500},
7463   "3B = {200,400},
7464   "3C = {200,100},
7465   "3D = {200,200},
7466   "3E = {100,200},
7467   A = { ,100},
7468   D = { , 50},
7469   J = { 50, },
7470   K = { , 50},
7471   L = { , 50},
7472   Q = { , 50},
7473   T = { 50, },
7474   X = { 50, 50},
7475   Y = { 50, },
7476   h = { , 50},
7477   k = { , 50}
7478 }
7479

```

Extended by the eulervm package.

```

7480 \SetProtrusion
7481 [ name = euler-vm,
7482   load = euler ]
7483 { encoding = U,
7484   family = zeur }

```

```

7485 {
7486 "28 = {100,200},
7487 "29 = {100,200},
7488 "2A = {100,150},
7489 "2B = {100,150},
7490 "2C = {200,300},
7491 "2D = {200,300},
7492 "2E = { ,100},
7493 "2F = {100, },
7494 "3F = {150,150},
7495 "5B = { ,100},
7496 "5E = {100,100},
7497 "5F = {100,100},
7498 "80 = { , 50},
7499 "81 = {200,250},
7500 "82 = {100,200}
7501 }
7502
7503 (/eur)

```

### Euler Script font (euca1).

```

7504 (*eus)
7505 \SetProtrusion
7506 [ name = euscript ]
7507 { encoding = U,
7508 family = eus }
7509 {
7510 A = {100,100},
7511 B = { 50,100},
7512 C = { 50, 50},
7513 D = { 50,100},
7514 E = { 50,100},
7515 F = { 50, },
7516 G = { 50, },
7517 H = { ,100},
7518 K = { , 50},
7519 L = { ,150},
7520 M = { , 50},
7521 N = { , 50},
7522 O = { 50, 50},
7523 P = { 50, 50},
7524 T = { ,100},
7525 U = { , 50},
7526 V = { 50, 50},
7527 W = { 50, 50},
7528 X = { 50, 50},
7529 Y = { 50, },
7530 Z = { 50,100},
7531 "00 = {250,250},
7532 "18 = {200,200},
7533 "3A = {200,150},
7534 "40 = { ,100},
7535 "5E = {100,100},
7536 "5F = {100,100},
7537 "66 = { 50, },
7538 "67 = { , 50},
7539 "6E = {200,200}
7540 }
7541
7542 \SetProtrusion
7543 [ name = euscript-vm,
7544 load = euscript ]
7545 { encoding = U,
7546 family = zeus }
7547 {

```

```
7548 "01 = {600,600},
7549 "02 = {200,200},
7550 "03 = {200,200},
7551 "04 = {200,200},
7552 "05 = {150,150},
7553 "06 = {200,200},
7554 "07 = {200,200},
7555 "08 = {100,100},
7556 "09 = {100,100},
7557 "0A = {100,100},
7558 "0B = {100,100},
7559 "0C = {100,100},
7560 "0D = {100,100},
7561 "0E = {150,150},
7562 "0F = {100,100},
7563 "10 = {150,150},
7564 "11 = {100,100},
7565 "12 = {150,100},
7566 "13 = {100,150},
7567 "14 = {150,100},
7568 "15 = {100,150},
7569 "16 = {200,100},
7570 "17 = {100,200},
7571 "19 = {150,150},
7572 "1A = {150,100},
7573 "1B = {100,150},
7574 "1C = {100,100},
7575 "1D = {100,100},
7576 "1E = {250,100},
7577 "1F = {100,250},
7578 "20 = {150,200},
7579 "21 = {150,200},
7580 "22 = {150,150},
7581 "23 = {150,150},
7582 "24 = {100,200},
7583 "25 = {150,150},
7584 "26 = {150,150},
7585 "27 = {100,100},
7586 "28 = {100,100},
7587 "29 = {100,150},
7588 "2A = {100,100},
7589 "2B = {100,100},
7590 "2C = {100,100},
7591 "2D = {150,150},
7592 "2E = {150,150},
7593 "2F = {100,100},
7594 "30 = {100,100},
7595 "31 = {100,100},
7596 "32 = {100,100},
7597 "33 = {100,100},
7598 "34 = {100,100},
7599 "35 = {100,100},
7600 "3E = {150,150},
7601 "3F = {150,150},
7602 "60 = { ,200},
7603 "61 = {200, },
7604 "62 = {100,100},
7605 "63 = {100,100},
7606 "64 = {100,100},
7607 "65 = {100,100},
7608 "68 = {300, },
7609 "69 = { ,300},
7610 "6C = {100,100},
7611 "6D = {100,100},
7612 "6F = {100,100},
```

```

7613     "72 = {100,100},
7614     "73 = {200,100},
7615     "76 = {   ,100},
7616     "77 = {100,   },
7617     "78 = { 50, 50},
7618     "79 = {100,100},
7619     "7A = {100,100},
7620     "7D = {150,150},
7621     "7E = {100,100},
7622     "A8 = {100,100},
7623     "A9 = {100,100},
7624     "AB = {200,200},
7625     "BA = {   ,200},
7626     "BB = {   ,200},
7627     "BD = {200,200},
7628     "DE = {200,200}
7629   }
7630
7631 (/eus)

```

### Euler Fraktur font (eufrak).

```

7632 (*euf)
7633 \SetProtrusion
7634   [ name      = mathfrak ]
7635   { encoding = U,
7636     family   = euf   }
7637   {
7638     A = {   , 50},
7639     B = {   , 50},
7640     C = { 50, 50},
7641     D = {   , 80},
7642     E = { 50,   },
7643     G = {   , 50},
7644     L = {   , 80},
7645     O = {   , 50},
7646     T = {   , 80},
7647     X = { 80, 50},
7648     Z = { 80, 50},
7649     b = {   , 50},
7650     c = {   , 50},
7651     k = {   , 50},
7652     p = {   , 50},
7653     q = { 50,   },
7654     v = {   , 50},
7655     w = {   , 50},
7656     x = {   , 50},
7657     1 = {100,100},
7658     2 = { 80, 80},
7659     3 = { 80, 50},
7660     4 = { 80, 50},
7661     7 = { 50, 50},
7662     "12 = {500,500},
7663     "13 = {500,500},
7664     ! = {   ,200},
7665     ' = {200,300},
7666     ( = {200,   },
7667     ) = {   ,200},
7668     * = {200,200},
7669     + = {200,250},
7670     - = {200,200},
7671     {,} = {300,300},
7672     . = {400,400},
7673     {=} = {200,200},
7674     : = {   ,200},
7675     ; = {   ,200},

```

```

7676     ] = { ,200}
7677   }
7678
7679 </euf>
7680 </cfg-u>

```

### 15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, euros), ITC Euro fonts (package euroitc) and marvosym<sup>23</sup>).

```

7681 <*cfg-e>
7682 \SetProtrusion
7683 <zpeu|euroitc> { encoding = U,
7684 <mvs> { encoding = {OT1,U},
7685 <zpeu> family = zpeu }
7686 <euroitc> family = {euroitc,euroitcs} }
7687 <mvs> family = mvs }
7688 {
7689 <zpeu> E = {50, }
7690 <euroitc> E = {100,50}
7691 <mvs> 164 = {50,50}, % \EUR
7692 <mvs> 068 = {50,-100} % \EURdig
7693 }
7694
7695 <*zpeu|euroitc>
7696 \SetProtrusion
7697 { encoding = U,
7698 <zpeu> family = zpeu,
7699 <euroitc> family = {euroitc,euroitcs},
7700 shape = it* }
7701 {
7702 <zpeu> E = {100,-50}
7703 <euroitc> E = {100,}
7704 }
7705
7706 </zpeu|euroitc>
7707 <*zpeu>
7708 \SetProtrusion
7709 { encoding = U,
7710 family = {zpeus,eurosans} }
7711 {
7712 E = {100,50}
7713 }
7714
7715 \SetProtrusion
7716 { encoding = U,
7717 family = {zpeus,eurosans},
7718 shape = it* }
7719 {
7720 E = {200, }
7721 }
7722
7723 </zpeu>
7724 </cfg-e>

```

## 15.9 Interword spacing

Default unit is space.

```

7725 <*m-t|cmr>
7726 %%% -----

```

---

23 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.

2      6      7    5      3            4            1

Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei
Das	Aus	kam	in	der	letzten	Runde,	wobei

```

7727 %% INTERWORD SPACING
7728
7729 </m-t|cmr>
7730 <+m-t>
7731 \SetExtraSpacing
7732   [ name = default ]
7733   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7734   {

```

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
- 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)

```
7735   { , } = { , -500, 500 },
```

```
7736   r = { , -300, 300 },
```

- [before or] after lowercase characters with ascenders

```
7737   b = { , -200, 200 },
```

```
7738   d = { , -200, 200 },
```

```
7739   f = { , -200, 200 },
```

```
7740   h = { , -200, 200 },
```

```
7741   k = { , -200, 200 },
```

```
7742   l = { , -200, 200 },
```

```
7743   t = { , -200, 200 },
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., ‘v’, or ‘w’

```
7744   c = { , -100, 100 },
```

```
7745   p = { , -100, 100 },
```

```
7746   v = { , -100, 100 },
```

```
7747   w = { , -100, 100 },
```

```
7748   z = { , -100, 100 },
```

```
7749   x = { , -100, 100 },
```

```
7750   y = { , -100, 100 },
```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```
7751         i = { , 50, -50},
7752         m = { , 50, -50},
7753         n = { , 50, -50},
7754         u = { , 50, -50},
```

- after colon and semicolon

```
7755         : = { ,200,-200},
7756         ; = { ,200,-200},
```

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

```
7757         . = { ,250,-250},
7758         ! = { ,250,-250},
7759         ? = { ,250,-250}
```

The order has to be reversed when enlarging is needed.’

```
7760     }
7761
7762 </m-t>
```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor < 1000.)
- Should the first parameter also be used? (Probably.)
- What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTeX bug with spacing in combination with a non-zero `\spaceskip` (reported by *Axel Berger*):

```
\parfillskip0pt
\righskip0pt plus 1em
\spaceskip\fontdimen2\font
test test\par
\pdfadjustinterwordglue2
\stbscode\font`t=-50
test test
\bye
```

Some more characters in T2A.<sup>24</sup>

```
7763 <+cmr>
7764 \SetExtraSpacing
7765 [ name = T2A,
7766   load = default ]
7767 { encoding = T2A,
7768   family = cmr }
7769 {
7770   \cyrg = { ,-300,300},
7771   \cyrb = { ,-200,200},
7772   \cyrk = { ,-200,200},
7773   \cyrs = { ,-100,100},
7774   \cyrr = { ,-100,100},
7775   \cyrh = { ,-100,100},
7776   \cyru = { ,-100,100},
7777   \cyrt = { , 50, -50},
7778   \cyrp = { , 50, -50},
7779   \cyri = { , 50, -50},
```

```

7780   \cyrishrt = { , 50, -50},
7781   }
7782

```

### 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 `TEXbook`:

‘If the space factor  $f$  is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if  $f \geq 2000$ . [...] Then the stretch component is multiplied by  $f/1000$ , while the shrink component is multiplied by  $1000/f$ .’

The ‘extra space’ (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i.e., 333.

```

7783 \SetExtraSpacing
7784   [ name = nonfrench-cmr,
7785     load = default,
7786     context = nonfrench ]
7787   { encoding = {OT1,T1,LY1,OT4,QX,T5},
7788     family = cmr }
7789   {

```

`latex.ltx` has:

```

\def\nonfrenchspacing{
  \sfcode\ . 3000
  \sfcode\? 3000
  \sfcode\! 3000

```

```

7790   . = {333,2000,-667},
7791   ? = {333,2000,-667},
7792   ! = {333,2000,-667},

```

```

\sfcode\ : 2000

```

```

7793   : = {333,1000,-500},

```

```

\sfcode\ ; 1500

```

```

7794   ; = { , 500,-333},

```

```

\sfcode\ , 1250

```

```

7795   {,}= { , 250,-200}

```

```

}

```

```

7796   }
7797
7798 </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.

```

7799 < *m-t >
7800 \SetExtraSpacing

```

```

7801 [ name      = nonfrench-default,
7802   load      = default,
7803   context   = nonfrench ]
7804 { encoding = {OT1,T1,LY1,OT4,QX,T5} }
7805 {
7806   . = {240,2000,-667},
7807   ? = {240,2000,-667},
7808   ! = {240,2000,-667},
7809   : = {240,1000,-500},
7810   ; = {   , 500,-333},
7811   {,}= {   , 250,-200}
7812 }
7813

```

## 15.10 Additional kerning

Default unit is 1 em.

```

7814 %%% -----
7815 %%% ADDITIONAL KERNING
7816

```

A dummy list to be loaded when no context is active.

```

7817 \SetExtraKerning
7818 [ name = empty ]
7819 { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
7820 { }
7821

```

### 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>25</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.)

```

7822 \SetExtraKerning
7823 [ name      = french-default,
7824   context   = french,
7825   unit      = space ]
7826 { encoding = {OT1,T1,LY1} }
7827 {
7828   : = {1000,}, % = \fontdimen2
7829   ; = {500, }, % ~ \thinspace
7830   ! = {500, },
7831   ? = {500, }
7832 }
7833

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTeX.

```

7834 \SetExtraKerning
7835 [ name      = french-guillemets,
7836   context   = french-guillemets,
7837   load      = french-default,
7838   unit      = space ]

```

---

25 [http://fr.wikipedia.org/wiki/Espace\\_typographique](http://fr.wikipedia.org/wiki/Espace_typographique), 5 July 2007.

```
7839 { encoding = {T1,LY1} }
7840 {
7841   \guillemotleft = { ,800}, % = 0.8\fontdimen2
7842   \guillemotright = {800, }
7843 }
7844
7845 \SetExtraKerning
7846 [ name = french-guillemets-OT1,
7847   context = french-guillemets,
7848   load = french-default,
7849   unit = space ]
7850 { encoding = OT1 }
7851 { }
7852
```

### 15.10.2 Turkish

```
7853 \SetExtraKerning
7854 [ name = turkish,
7855   context = turkish ]
7856 { encoding = {OT1,T1,LY1} }
7857 {
7858   : = {167, }, % = \thinspace
7859   ! = {167, },
7860   {=} = {167, }
7861 }
7862
7863 /m-t
7864 /config
```





7970 T = {T, T̄, T̂, T̃, 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, 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}, % Cyr  
7971 T, T̄}, % Cyr  
7972 U = {Ū, Ū̄, Ū̂, Ū̃, Ū̄, Ū̅, Ū̆, Ū̇, Ṻ, Ū̉, Ū̊, Ū̋, Ū̌, Ū̍, Ū̎, Ū̏, Ū̐, Ū̑, Ū̒, Ū̓, Ū̔, Ū̕, Ū̖, Ū̗, Ū̘, Ū̙, Ū̚, Ư̄, Ū̜, Ū̝, Ū̞, Ū̟, Ū̠, Ū̡, Ū̢, Ụ̄, Ṳ̄, Ū̥, Ū̦, Ū̧, Ų̄, Ū̩, Ū̪, Ū̫, Ū̬, Ṷ̄, Ū̮, Ū̯, Ṵ̄, Ū̱, Ū̲, Ū̳, Ū̴, Ū̵, Ū̶, Ū̷, Ū̸, Ū̹, Ū̺, Ū̻, Ū̼, Ū̽, Ū̾, Ū̿}, % Cyr  
7973 V = {V̄, V̅},  
7974 W = {Ŵ, Ŵ̄, Ŵ̂, Ŵ̃, 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}, % Cyr  
7975 W}, % Cyr  
7976 X = {X̄, X̅,  
7977 X̆, Ẋ, Ẍ, X̉, X̊, X̋, X̌, X̍, X̎, X̏, X̐, X̑, X̒, X̓, X̔, X̕, X̖, X̗, X̘, X̙, X̚, X̛, X̜, X̝, X̞, X̟, X̠, X̡, X̢, X̣, X̤, X̥, X̦, X̧, X̨, X̩, X̪, X̫, X̬, X̭, X̮, X̯, X̰, X̱, X̲, X̳, X̴, X̵, X̶, X̷, X̸, X̹, X̺, X̻, X̼, X̽, X̾, X̿}, % Cyr  
7978 Y = {Ÿ, Ÿ̄, Ÿ̂, Ÿ̃, Ÿ̄, Ÿ̅, Ÿ̆, Ÿ̇, Ÿ̈, Ÿ̉, Ÿ̊, Ÿ̋, Ÿ̌, Ÿ̍, Ÿ̎, Ÿ̏, Ÿ̐, Ÿ̑, Ÿ̒, Ÿ̓, Ÿ̔, Ÿ̕, Ÿ̖, Ÿ̗, Ÿ̘, Ÿ̙, Ÿ̚, Ÿ̛, Ÿ̜, Ÿ̝, Ÿ̞, Ÿ̟, Ÿ̠, Ÿ̡, Ÿ̢, Ỵ̈, Ÿ̤, Ÿ̥, Ÿ̦, Ÿ̧, Ÿ̨, Ÿ̩, Ÿ̪, Ÿ̫, Ÿ̬, Ÿ̭, Ÿ̮, Ÿ̯, Ÿ̰, Ÿ̱, Ÿ̲, Ÿ̳, Ÿ̴, Ÿ̵, Ÿ̶, Ÿ̷, Ÿ̸, Ÿ̹, Ÿ̺, Ÿ̻, Ÿ̼, Ÿ̽, Ÿ̾, Ÿ̿}, % Cyr  
7979 Y, Ÿ}, % Cyr  
7980 Z = {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}, % a'  
7981 a = {Ǻ, Ǻ̄, Ǻ̂, Ǻ̃, Ǻ̄, Ǻ̅, Ǻ̆, Ǻ̇, Ǻ̈, Ǻ̉, Ǻ̊, Ǻ̋, Ǻ̌, Ǻ̍, Ǻ̎, Ǻ̏, Ǻ̐, Ǻ̑, Ǻ̒, Ǻ̓, Ǻ̔, Ǻ̕, Ǻ̖, Ǻ̗, Ǻ̘, Ǻ̙, Ǻ̚, Ǻ̛, Ǻ̜, Ǻ̝, Ǻ̞, Ǻ̟, Ǻ̠, Ǻ̡, Ǻ̢, Ạ̊́, Ǻ̤, Ḁ̊́, Ǻ̦, Ǻ̧, Ą̊́, Ǻ̩, Ǻ̪, Ǻ̫, Ǻ̬, Ǻ̭, Ǻ̮, Ǻ̯, Ǻ̰, Ǻ̱, Ǻ̲, Ǻ̳, Ǻ̴, Ǻ̵, Ǻ̶, Ǻ̷, Ǻ̸, Ǻ̹, Ǻ̺, Ǻ̻, Ǻ̼, Ǻ̽, Ǻ̾, Ǻ̿}, % a'  
7982 a, Ǻ̄}, % Cyr  
7983 æ = {Ǽ,  
7984 æ}, % Cyr  
7985 b = {b̄, b̅, b̆},  
7986 c = {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}, % Cyr  
7987 c, ç}, % Cyr  
7988 d = {d̄, d̅, d̆, ḋ, d̈, d̉, d̊, d̋, ď, d̍, d̎, d̏, d̐, d̑, d̒, d̓, d̔, d̕, d̖, d̗, d̘, d̙, d̚, d̛, d̜, d̝, d̞, d̟, d̠, d̡, d̢, ḍ, d̤, d̥, d̦, ḑ, d̨, d̩, d̪, d̫, d̬, ḓ, d̮, d̯, d̰, ḏ, d̲, d̳, d̴, d̵, d̶, d̷, d̸, d̹, d̺, d̻, d̼, d̽, d̾, d̿}, % Cyr  
7989 e = {è, è̄, è̂, è̃, 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}, % Cyr  
7990 e, è̄, è̅}, % Cyr  
7991 f = {f̄, f̅, f̆}, % /f f  
7992 g = {ğ̄, ğ̅, ğ̆, ğ̇, ğ̈, ğ̉, ğ̊, ğ̋, ğ̌, ğ̍, ğ̎, ğ̏, ğ̐, ğ̑, ğ̒, ğ̓, ğ̔, ğ̕, ğ̖, ğ̗, ğ̘, ğ̙, ğ̚, ğ̛, ğ̜, ğ̝, ğ̞, ğ̟, ğ̠, ğ̡, ğ̢, ğ̣, ğ̤, ğ̥, ğ̦, ģ̆, ğ̨, ğ̩, ğ̪, ğ̫, ğ̬, ğ̭, ğ̮, ğ̯, ğ̰, ğ̱, ğ̲, ğ̳, ğ̴, ğ̵, ğ̶, ğ̷, ğ̸, ğ̹, ğ̺, ğ̻, ğ̼, ğ̽, ğ̾, ğ̿}, % Cyr  
7993 h = {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}, % Cyr  
7994 h, h̄}, % Cyr  
7995 i = {ī, i̅, ĭ, i̇, ï, ỉ, i̊, i̋, ǐ, i̍, i̎, ȉ, i̐, ȋ, i̒, i̓, i̔, i̕, i̖, i̗, i̘, i̙, i̚, i̛, i̜, i̝, i̞, i̟, i̠, i̡, i̢, ị, i̤, i̥, i̦, i̧, į, i̩, i̪, i̫, i̬, i̭, i̮, i̯, ḭ, i̱, i̲, i̳, i̴, i̵, i̶, i̷, i̸, i̹, i̺, i̻, i̼, i̽, i̾, i̿}, % Cyr  
7996 i, ī}, % Cyr  
7997 j = {j̄, j̅, j̆}, % Cyr  
7998 j}, % Cyr  
7999 k = {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}, % Cyr  
8000 l = {l̄, l̅, l̆, l̇, l̈, l̉, l̊, l̋, ľ, l̍, l̎, l̏, l̐, l̑, l̒, l̓, l̔, l̕, l̖, l̗, l̘, l̙, l̚, l̛, l̜, l̝, l̞, l̟, l̠, l̡, l̢, ḷ, l̤, l̥, l̦, ļ, l̨, l̩, l̪, l̫, l̬, ḽ, l̮, l̯, l̰, ḻ, l̲, l̳, l̴, l̵, l̶, l̷, l̸, l̹, l̺, l̻, l̼, l̽, l̾, l̿}, % l, l̄  
8001 m = {m̄, m̅, m̆},  
8002 n = {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}, % 'n  
8003 o = {ò, ò̄, ò̂, ò̃, ò̄, ò̅, ò̆, ò̇, ò̈, ò̉, ò̊, ò̋, ò̌, ò̍, ò̎, ò̏, ò̐, ò̑, ò̒, ò̓, ò̔, ò̕, ò̖, ò̗, ò̘, ò̙, ò̚, ờ, ò̜, ò̝, ò̞, ò̟, ò̠, ò̡, ò̢, ọ̀, ò̤, ò̥, ò̦, ò̧, ǫ̀, ò̩, ò̪, ò̫, ò̬, ò̭, ò̮, ò̯, ò̰, ò̱, ò̲, ò̳, ò̴, ò̵, ò̶, ò̷, ò̸, ò̹, ò̺, ò̻, ò̼, ò̽, ò̾, ò̿}, % Cyr  
8004 o, ò̄, ò̅}, % Cyr  
8005 p = {p̄, p̅, p̆}, % Cyr  
8006 p, p̄}, % Cyr  
8007 q = {q̄, q̅, q̆}, % Cyr  
8008 r = {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}, % Cyr  
8009 s = {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}, % Cyr  
8010 s}, % Cyr  
8011 t = {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}, % t  
8012 u = {ù, ù̄, ù̂, ù̃, 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}, % Cyr  
8013 v = {v̄, v̅},  
8014 w = {ŵ, ŵ̄, ŵ̂, ŵ̃, 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}, % Cyr  
8015 w}, % Cyr  
8016 x = {x̄, x̅,  
8017 x̆, ẋ, ẍ, x̉, x̊, x̋, x̌, x̍, x̎, x̏, x̐, x̑, x̒, x̓, x̔, x̕, x̖, x̗, x̘, x̙, x̚, x̛, x̜, x̝, x̞, x̟, x̠, x̡, x̢, x̣, x̤, x̥, x̦, x̧, x̨, x̩, x̪, x̫, x̬, x̭, x̮, x̯, x̰, x̱, x̲, x̳, x̴, x̵, x̶, x̷, x̸, x̹, x̺, x̻, x̼, x̽, x̾, x̿}, % Cyr  
8018 y = {ý, ý̄, ý̂, ý̃, 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}, % Cyr  
8019 y, ý̄, ý̅}, % Cyr  
8020 z = {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}, % Cyrillic  
8021 % Cyrillic  
8022 Г = {Г̄, Г̅, Г̆, Г̇, Г̈, Г̉, Г̊, Г̋, Г̌, Г̍, Г̎, Г̏, Г̐, Г̑, Г̒, Г̓, Г̔, Г̕, Г̖, Г̗, Г̘, Г̙, Г̚, Г̛, Г̜, Г̝, Г̞, Г̟, Г̠, Г̡, Г̢, Г̣, Г̤, Г̥, Г̦, Г̧, Г̨, Г̩, Г̪, Г̫, Г̬, Г̭, Г̮, Г̯, Г̰, Г̱, Г̲, Г̳, Г̴, Г̵, Г̶, Г̷, Г̸, Г̹, Г̺, Г̻, Г̼, Г̽, Г̾, Г̿}, % Cyrillic  
8023 Ж = {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}, % Cyrillic  
8024 З = {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}, % Cyrillic  
8025 Л = {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}, % Cyrillic  
8026 П = {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}, % Cyrillic  
8027 Я = {Я̄, Я̅, Я̆, Я̇, Я̈, Я̉, Я̊, Я̋, Я̌, Я̍, Я̎, Я̏, Я̐, Я̑, Я̒, Я̓, Я̔, Я̕, Я̖, Я̗, Я̘, Я̙, Я̚, Я̛, Я̜, Я̝, Я̞, Я̟, Я̠, Я̡, Я̢, Я̣, Я̤, Я̥, Я̦, Я̧, Я̨, Я̩, Я̪, Я̫, Я̬, Я̭, Я̮, Я̯, Я̰, Я̱, Я̲, Я̳, Я̴, Я̵, Я̶, Я̷, Я̸, Я̹, Я̺, Я̻, Я̼, Я̽, Я̾, Я̿}, % Cyrillic  
8028 Ч = {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}, % Cyrillic  
8029 Ъ = {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}, % Cyrillic  
8030 Ы = {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}, % Cyrillic  
8031 Ь = {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}, % Cyrillic  
8032 Г = {Г̄, Г̅, Г̆, Г̇, Г̈, Г̉, Г̊, Г̋, Г̌, Г̍, Г̎, Г̏, Г̐, Г̑, Г̒, Г̓, Г̔, Г̕, Г̖, Г̗, Г̘, Г̙, Г̚, Г̛, Г̜, Г̝, Г̞, Г̟, Г̠, Г̡, Г̢, Г̣, Г̤, Г̥, Г̦, Г̧, Г̨, Г̩, Г̪, Г̫, Г̬, Г̭, Г̮, Г̯, Г̰, Г̱, Г̲, Г̳, Г̴, Г̵, Г̶, Г̷, Г̸, Г̹, Г̺, Г̻, Г̼, Г̽, Г̾, Г̿}, % Cyrillic  
8033 ж = {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}, % Cyrillic

```

8034   з = {з,з̄},
8035   и = {Ӏ,Ӏ̄,Ӏ̂,Ӏ̃,Ӏ̄},
8036   к = {к,к̄,к̂,к̃,к̄,к̅},
8037   л = {л},
8038   м = {м},
8039   н = {н,н̄,н̂,н̃},
8040   п = {п},
8041   т = {т},
8042   х = {х,х̄},
8043   ч = {ч,ч̄,ч̂},
8044   ш = {ш},
8045   ы = {ы̄},
8046   э = {э̄},
8047   ъ = {ъ},
8048   ӱ = {ӱ̄},
8049   γ = {γ},
8050   Γ = {Γ}, % Greek
8051   Π = {Π}, % Greek
8052 }
8053
8054 % missing: tipa, math, symbols, ...
8055 /CharisSIL
8056 *PalatinoLinotype
8057 \DeclareCharacterInheritance
8058 { encoding = {EU1,EU2,TU},
8059   family = {PalatinoLinotype} }

```

Unfortunately, I don't have a Palatino variant containing all of the following glyphs. The settings are typeset in TeX Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as '◆'. To see the real settings, consult `mt-PalatinoLinotype.cfg`.

```

8060 { A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ȧ,Ǽ,Ǽ̄,Ǽ̂,Ǽ̃,Ǽ̄,Ǽ̅,Ǽ̆,Ǽ̇,Ǽ̈,Ǽ̉,Ǽ̊,Ǽ̋,Ǽ̌,Ǽ̍,Ǽ̎,Ǽ̏,Ǽ̐,Ǽ̑,Ǽ̒,Ǽ̓,Ǽ̔,Ǽ̕,Ǽ̖,Ǽ̗,Ǽ̘,Ǽ̙,Ǽ̚,Ǽ̛,Ǽ̜,Ǽ̝,Ǽ̞,Ǽ̟,Ǽ̠,Ǽ̡,Ǽ̢,Ǽ̣,Ǽ̤,Ǽ̥,Ǽ̦,Ǽ̧,Ǽ̨,Ǽ̩,Ǽ̪,Ǽ̫,Ǽ̬,Ǽ̭,Ǽ̮,Ǽ̯,Ǽ̰,Ǽ̱,Ǽ̲,Ǽ̳,Ǽ̴,Ǽ̵,Ǽ̶,Ǽ̷,Ǽ̸,Ǽ̹,Ǽ̺,Ǽ̻,Ǽ̼,Ǽ̽,Ǽ̾,Ǽ̿,Ǽ̿},
8061   B = {B̂,B̃,B̄},
8062   C = {Ç,Ć,Ĉ,Ċ,Č,Ď},
8063   D = {D̂,D̃,D̄,D̅},
8064   E = {È,É,Ê,Ë,Ē,Ē̄,Ē̂,Ē̃,Ē̄,Ē̅,Ē̆,Ē̇,Ē̈,Ē̉,Ē̊,Ē̋,Ē̌,Ē̍,Ē̎,Ē̏,Ē̐,Ē̑,Ē̒,Ē̓,Ē̔,Ē̕,Ē̖,Ē̗,Ē̘,Ē̙,Ē̚,Ē̛,Ē̜,Ē̝,Ē̞,Ē̟,Ē̠,Ē̡,Ē̢,Ẹ̄,Ē̤,Ē̥,Ē̦,Ȩ̄,Ę̄,Ē̩,Ē̪,Ē̫,Ē̬,Ḙ̄,Ē̮,Ē̯,Ḛ̄,Ē̱,Ē̲,Ē̳,Ē̴,Ē̵,Ē̶,Ē̷,Ē̸,Ē̹,Ē̺,Ē̻,Ē̼,Ē̽,Ē̾,Ē̿,Ē̿},
8065   F = {F̂},
8066   G = {Ĝ,Ĝ̂,Ĝ̃,Ĝ̄,Ĝ̅},
8067   H = {Ĥ,Ĥ̂,Ĥ̃,Ĥ̄,Ĥ̅},
8068   I = {İ,İ̂,İ̃,İ̄,İ̅,İ̆,İ̇,İ̈,İ̉,İ̊,İ̋,İ̌,İ̍,İ̎,İ̏,İ̐,İ̑,İ̒,İ̓,İ̔,İ̕,İ̖,İ̗,İ̘,İ̙,İ̚,İ̛,İ̜,İ̝,İ̞,İ̟,İ̠,İ̡,İ̢,Ị̇,İ̤,İ̥,İ̦,İ̧,Į̇,İ̩,İ̪,İ̫,İ̬,İ̭,İ̮,İ̯,Ḭ̇,İ̱,İ̲,İ̳,İ̴,İ̵,İ̶,İ̷,İ̸,İ̹,İ̺,İ̻,İ̼,İ̽,İ̾,İ̿,İ̿},
8069   J = {Ĵ},
8070   K = {K̂,K̃,K̄,K̅},
8071   L = {Ĺ,Ĺ̂,Ĺ̃,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,Ĺ̿}, % L
8072   M = {M̂,M̃,M̄},
8073   N = {Ñ,Ñ̂,Ñ̃,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,Ñ̿},
8074   O = {Ò,Ó,Ô,Õ,Ö,Ï,Ï̄,Ï̂,Ï̃,Ï̄,Ï̅,Ï̆,Ï̇,Ï̈,Ï̉,Ï̊,Ï̋,Ï̌,Ï̍,Ï̎,Ï̏,Ï̐,Ï̑,Ï̒,Ï̓,Ï̔,Ï̕,Ï̖,Ï̗,Ï̘,Ï̙,Ï̚,Ï̛,Ï̜,Ï̝,Ï̞,Ï̟,Ï̠,Ï̡,Ï̢,Ị̈,Ï̤,Ï̥,Ï̦,Ï̧,Į̈,Ï̩,Ï̪,Ï̫,Ï̬,Ï̭,Ï̮,Ï̯,Ḭ̈,Ï̱,Ï̲,Ï̳,Ï̴,Ï̵,Ï̶,Ï̷,Ï̸,Ï̹,Ï̺,Ï̻,Ï̼,Ï̽,Ï̾,Ï̿,Ï̿},
8075   P = {P̂,P̃},
8076   R = {R̂,R̃,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,R̿},
8077   S = {Ś,Ś̂,Ś̃,Ś̄,Ś̅,Ś̆,Ṥ,Ś̈,Ś̉,Ś̊,Ś̋,Ś̌,Ś̍,Ś̎,Ś̏,Ś̐,Ś̑,Ś̒,Ś̓,Ś̔,Ś̕,Ś̖,Ś̗,Ś̘,Ś̙,Ś̚,Ś̛,Ś̜,Ś̝,Ś̞,Ś̟,Ś̠,Ś̡,Ś̢,Ṣ́,Ś̤,Ś̥,Ș́,Ş́,Ś̨,Ś̩,Ś̪,Ś̫,Ś̬,Ś̭,Ś̮,Ś̯,Ś̰,Ś̱,Ś̲,Ś̳,Ś̴,Ś̵,Ś̶,Ś̷,Ś̸,Ś̹,Ś̺,Ś̻,Ś̼,Ś̽,Ś̾,Ś̿,Ś̿},
8078   T = {T̂,T̃,T̄,T̅},
8079   U = {Û,Ū,Ū̂,Ū̃,Ū̄,Ū̅,Ū̆,Ū̇,Ṻ,Ū̉,Ū̊,Ū̋,Ū̌,Ū̍,Ū̎,Ū̏,Ū̐,Ū̑,Ū̒,Ū̓,Ū̔,Ū̕,Ū̖,Ū̗,Ū̘,Ū̙,Ū̚,Ư̄,Ū̜,Ū̝,Ū̞,Ū̟,Ū̠,Ū̡,Ū̢,Ụ̄,Ṳ̄,Ū̥,Ū̦,Ū̧,Ų̄,Ū̩,Ū̪,Ū̫,Ū̬,Ṷ̄,Ū̮,Ū̯,Ṵ̄,Ū̱,Ū̲,Ū̳,Ū̴,Ū̵,Ū̶,Ū̷,Ū̸,Ū̹,Ū̺,Ū̻,Ū̼,Ū̽,Ū̾,Ū̿,Ū̿},
8080   V = {V̂,Ṽ},
8081   W = {Ŵ,Ŵ̂,Ŵ̃,Ŵ̄,Ŵ̅},
8082   X = {X̂,X̃},
8083   Y = {Ŷ,Ŷ̂,Ŷ̃,Ŷ̄,Ŷ̅,Ŷ̆,Ŷ̇,Ŷ̈,Ŷ̉,Ŷ̊,Ŷ̋,Ŷ̌,Ŷ̍,Ŷ̎,Ŷ̏,Ŷ̐,Ŷ̑,Ŷ̒,Ŷ̓,Ŷ̔,Ŷ̕,Ŷ̖,Ŷ̗,Ŷ̘,Ŷ̙,Ŷ̚,Ŷ̛,Ŷ̜,Ŷ̝,Ŷ̞,Ŷ̟,Ŷ̠,Ŷ̡,Ŷ̢,Ỵ̂,Ŷ̤,Ŷ̥,Ŷ̦,Ŷ̧,Ŷ̨,Ŷ̩,Ŷ̪,Ŷ̫,Ŷ̬,Ŷ̭,Ŷ̮,Ŷ̯,Ŷ̰,Ŷ̱,Ŷ̲,Ŷ̳,Ŷ̴,Ŷ̵,Ŷ̶,Ŷ̷,Ŷ̸,Ŷ̹,Ŷ̺,Ŷ̻,Ŷ̼,Ŷ̽,Ŷ̾,Ŷ̿,Ŷ̿},
8084   Z = {Ž,Ž̂,Ž̃,Ž̄,Ž̅},
8085   a = {ā,ă,ą,ȧ,Ǽ,Ǽ̄,Ǽ̂,Ǽ̃,Ǽ̄,Ǽ̅,Ǽ̆,Ǽ̇,Ǽ̈,Ǽ̉,Ǽ̊,Ǽ̋,Ǽ̌,Ǽ̍,Ǽ̎,Ǽ̏,Ǽ̐,Ǽ̑,Ǽ̒,Ǽ̓,Ǽ̔,Ǽ̕,Ǽ̖,Ǽ̗,Ǽ̘,Ǽ̙,Ǽ̚,Ǽ̛,Ǽ̜,Ǽ̝,Ǽ̞,Ǽ̟,Ǽ̠,Ǽ̡,Ǽ̢,Ǽ̣,Ǽ̤,Ǽ̥,Ǽ̦,Ǽ̧,Ǽ̨,Ǽ̩,Ǽ̪,Ǽ̫,Ǽ̬,Ǽ̭,Ǽ̮,Ǽ̯,Ǽ̰,Ǽ̱,Ǽ̲,Ǽ̳,Ǽ̴,Ǽ̵,Ǽ̶,Ǽ̷,Ǽ̸,Ǽ̹,Ǽ̺,Ǽ̻,Ǽ̼,Ǽ̽,Ǽ̾,Ǽ̿,Ǽ̿}, % a^
8086   b = {b̂,b̃},
8087   c = {ç,ć,ĉ,ċ,č,ď},
8088   d = {đ,d̂,d̃,d̄,d̅},
8089   e = {è,é,ê,ë,ē,ē̄,ē̂,ē̃,ē̄,ē̅,ē̆,ē̇,ē̈,ē̉,ē̊,ē̋,ē̌,ē̍,ē̎,ē̏,ē̐,ē̑,ē̒,ē̓,ē̔,ē̕,ē̖,ē̗,ē̘,ē̙,ē̚,ē̛,ē̜,ē̝,ē̞,ē̟,ē̠,ē̡,ē̢,ẹ̄,ē̤,ē̥,ē̦,ȩ̄,ę̄,ē̩,ē̪,ē̫,ē̬,ḙ̄,ē̮,ē̯,ḛ̄,ē̱,ē̲,ē̳,ē̴,ē̵,ē̶,ē̷,ē̸,ē̹,ē̺,ē̻,ē̼,ē̽,ē̾,ē̿,ē̿},
8090   f = {f,ff},

```

```

8091 g = {ǧ,ǧ̇,ǧ̈,ǧ̉,ǧ̊,ǧ̋},
8092 h = {ĥ,ĥ̇,ĥ̈,ĥ̉,ĥ̊,ĥ̋},
8093 i = {î,î̇,î̈,î̉,î̊,î̋,ï,ï̇,ï̈,ï̉,ï̊,ï̋},
8094 j = {ĵ,ĵ̇},
8095 k = {ķ,ķ̇,ķ̈,ķ̉,ķ̊},
8096 l = {ĺ,ĺ̇,ĺ̈,ĺ̉,ĺ̊}, % ḷ.
8097 m = {ṁ,ṁ,m̈},
8098 n = {ñ,ñ̇,ñ̈,ñ̉,ñ̊,ñ̋}, % ñ.
8099 o = {ò,ò̇,ò̈,ò̉,ò̊,ò̋,ō,ō̇,ō̈,ō̉,ō̊,ō̋,ö,ö̇,ö̈,ö̉,ö̊,ö̋,ø,ø̇,ø̈,ø̉,ø̊,ø̋},
8100 p = {ṗ,ṗ},
8101 r = {ŕ,ŕ̇,ŕ̈,ŕ̉,ŕ̊,ŕ̋},
8102 s = {š,ṧ,š̈,š̉,š̊,š̋,ŝ,ŝ̇,ŝ̈,ŝ̉,ŝ̊,ŝ̋},
8103 t = {ţ,ţ̇,ţ̈,ţ̉,ţ̊}, % ṭ.
8104 u = {ù,ù̇,ù̈,ù̉,ù̊,ù̋,û,û̇,û̈,û̉,û̊,û̋,ü,ü̇,ü̈,ü̉,ü̊,ü̋,ū,ū̇,ṻ,ū̉,ū̊,ū̋,ǔ,ǔ̇,ǔ̈,ǔ̉,ǔ̊,ǔ̋},
8105 v = {ṽ,v̇},
8106 w = {ŵ,ẇ,ẅ,w̉,ẘ,w̋},
8107 x = {ẋ,ẋ},
8108 y = {ÿ,ÿ̇,ÿ̈,ÿ̉,ÿ̊,ÿ̋,ÿ̌,ÿ̌̇,ÿ̌̈,ÿ̌̉,ÿ̌̊,ÿ̌̋},
8109 z = {ž,ž̇,ž̈,ž̉,ž̊,ž̋},
8110 }
8111 (/PalatinoLinotype)

```

## 16.2 Character protrusion

```

8112
8113 %%% -----
8114 %%% PROTRUSION
8115
8116 (*LatinModernRoman)
8117 \SetProtrusion
8118 [ name = LMR-default ]
8119 { encoding = {EU1,EU2,TU},
8120 family = Latin Modern Roman }
8121 {
8122 A = {50,50},
8123 Æ = {50, },
8124 F = { ,50},
8125 J = {50, },
8126 K = { ,50},
8127 L = { ,50},
8128 T = {50,50},
8129 V = {50,50},
8130 W = {50,50},
8131 X = {50,50},
8132 Y = {50,50},
8133 k = { ,50},
8134 r = { ,50},
8135 t = { ,70},
8136 v = {50,50},
8137 w = {50,50},
8138 x = {50,50},
8139 y = {50,70},
8140 0 = { ,50},
8141 1 = {100,200},
8142 2 = {50,50},
8143 3 = {50,50},
8144 4 = {70,70},
8145 5 = { ,50},
8146 6 = { ,50},
8147 7 = {50,100},
8148 8 = { ,50},
8149 9 = { ,50},
8150 . = { ,700},

```

8151 {,}= { ,500},  
8152 := { ,500},  
8153 ;= { ,500},  
8154 != { ,100},  
8155 ? = { ,200},  
8156 @ = {50,50},  
8157 ~ = {200,250},  
8158 \% = {50,50},  
8159 \* = {300,300},  
8160 + = {250,250},  
8161 - = {400,500}, % /hyphen  
8162 – = {400,300}, % /endash  
8163 — = {300,200}, % /emdash  
8164 \_ = {200,200}, % /underscore  
8165 / = {200,300},  
8166 /backslash = {200,300},  
8167 ' = {300,400}, % /quotesingle  
8168 ‘ = {500,700}, ’ = {500,600},  
8169 “ = {500,300}, ” = {200,600},  
8170 , = {400,400}, ,, = {400,400},  
8171 ‹ = {400,400}, › = {300,500},  
8172 « = {300,200}, » = {100,400},  
8173 ¡ = {100, }, ¡ = {100, },  
8174 ( = {300, }, ) = { ,300},  
8175 < = {200,100}, > = {100,200},  
8176 /braceleft = {400,200}, /braceright = {200,400},  
8177 /angleleft = {400, }, /angleright = { ,400},  
8178 † = {100,100},  
8179 ‡ = { 80, 80},  
8180 • = {200,200},  
8181 · = {400,450}, % / periodcentered  
8182 °C = { 80, 50},  
8183 ℄ = { , 50},  
8184 ° = {400,400},  
8185 ™ = {100,200},  
8186 © = {100,100},  
8187 ® = {100,100},  
8188 ª = {100,200},  
8189 º = {100,200},  
8190 ¹ = {200,250},  
8191 º = { 50,100},  
8192 ³ = { 50,100},  
8193 ¬ = {200, },  
8194 − = {300,300},  
8195 ± = {150,200},  
8196 × = {150,250},  
8197 ÷ = {150,250},  
8198 € = {100, },  
8199 /one.oldstyle = {100,100},  
8200 /two.oldstyle = { 50, 50},  
8201 /three.oldstyle = { 30, 80},  
8202 /four.oldstyle = { 50, 50},  
8203 /seven.oldstyle = { 50, 80},  
8204 Γ = { ,180}, % /Gamma  
8205 Δ = {100,100}, % /Delta  
8206 Θ = { 50, 50}, % /Theta  
8207 Λ = {100,100}, % /Lambda  
8208 % Ξ = {,}, % /Xi  
8209 % Π = {,}, % /Pi  
8210 Σ = { 50, 50}, % /Sigma  
8211 Υ = {100,100}, % /Upsilon  
8212 Φ = { 50, 50}, % /Phi  
8213 Ψ = { 50, 50}, % /Psi  
8214 % Ω = {,}, % /Omega  
8215 }

```
8216
8217 \SetProtrusion
8218   [ name      = LMR-it ]
8219   { encoding = {EU1,EU2,TU},
8220     family   = Latin Modern Roman,
8221     shape    = {it,s1}      }
8222   {
8223     A = {125,100},
8224     Æ = {125,-55},
8225     B = {90,-40},
8226     C = {145,-75},
8227     D = {75, -28},
8228     E = {80,-55},
8229     F = {85,-80},
8230     G = {153,-15},
8231     H = {73,-60},
8232     I = {140,-120},
8233     IJ = {140,-80},
8234     J = {135,-80},
8235     K = {70,-30},
8236     L = {87, 40},
8237     M = {67,-45},
8238     N = {75,-55},
8239     O = {150,-30},
8240     Œ = {150,-55},
8241     P = {82,-50},
8242     Q = {150,-30},
8243     R = {75, 15},
8244     S = {90,-65},
8245     $ = {100,-20},
8246     T = {220,-85},
8247     U = {230,-55},
8248     V = {260,-60},
8249     W = {185,-55},
8250     X = {70,-30},
8251     Y = {250,-60},
8252     Z = {90,-60},
8253     a = {150,-10},
8254     b = {170,  },
8255     c = {173,-10},
8256     d = {150,-55},
8257     e = {180, },
8258     f = {  ,-250},
8259     g = {150,-10},
8260     h = {100, },
8261     i = {210, },
8262     ij = {210,-40},
8263     j = {  ,-40},
8264     k = {110,-50},
8265     l = {240,-110},
8266     m = {80, },
8267     n = {115, },
8268     o = {155, },
8269     q = {170,-40},
8270     r = {155,-40},
8271     s = {130, },
8272     t = {230,-10},
8273     u = {120, },
8274     v = {140,-25},
8275     w = {98,-20},
8276     x = {65,-40},
8277     y = {130,-20},
8278     z = {110,-80},
8279     0 = {170,-85},
8280     1 = {230,110},
```

8281 2 = {130,-70},  
 8282 3 = {140,-70},  
 8283 4 = {130,80},  
 8284 5 = {160, },  
 8285 6 = {175,-30},  
 8286 7 = {250,-150},  
 8287 8 = {130,-40},  
 8288 9 = {155,-80},  
 8289 . = { ,500},  
 8290 {,}= { ,450},  
 8291 : = { ,300},  
 8292 ; = { ,300},  
 8293 & = {130,30},  
 8294 \% = {180,50},  
 8295 \* = {380,20},  
 8296 + = {180,200},  
 8297 @ = {180,10},  
 8298 ~ = {200,150},  
 8299 ( = {300, }, ) = { ,70},  
 8300 / = {100,100},  
 8301 - = {500,300}, % /hyphen  
 8302 – = {500,300}, % /endash  
 8303 — = {400,170}, % /emdash  
 8304 \_ = {100,200}, % /underscore  
 8305 ' = {300,400}, % /quotesingle  
 8306 " = {500,300},  
 8307 ‘ = {800,200}, ’ = {800,-20},  
 8308 “ = {540,100}, ” = {500,100},  
 8309 , = {300,700}, ,, = {200,600},  
 8310 ‹ = {500,300}, › = {400,400},  
 8311 « = {400,100}, » = {200,300},  
 8312 ¡ = {200, }, ì = {200, },  
 8313 < = {300,100}, > = {200,100},  
 8314 /backslash = {300,300},  
 8315 /braceleft = {400,100}, /braceright = {200,200},  
 8316 † = {200, 80},  
 8317 ‡ = {120, 80},  
 8318 • = {220,100},  
 8319 · = {550,300}, % / periodcentered  
 8320 °C = {170, },  
 8321 © = {100, 50},  
 8322 ¶ = {200, },  
 8323 ° = {500,300},  
 8324 ™ = {200, 70},  
 8325 © = { 50, 70},  
 8326 ® = { 50, 70},  
 8327 ª = {140,100},  
 8328 º = {140,100},  
 8329 ¹ = {400,150},  
 8330 ² = {250, 80},  
 8331 ³ = {250, 80},  
 8332 ¬ = {250, 80},  
 8333 − = {300,200},  
 8334 ± = {150,170},  
 8335 × = {200,200},  
 8336 ÷ = {200,200},  
 8337 € = {150, },  
 8338 /one.oldstyle = {100,100},  
 8339 /two.oldstyle = {100, 80},  
 8340 /three.oldstyle = { 80, 50},  
 8341 /four.oldstyle = { 80, 80},  
 8342 /five.oldstyle = { 50, },  
 8343 /six.oldstyle = { 50, },  
 8344 /seven.oldstyle = { 80, 80},  
 8345 /eight.oldstyle = { 50, },

```

8346   Γ = {100,120}, % /Gamma
8347   Δ = {120,100}, % /Delta
8348   Θ = {120, 50}, % /Theta
8349   Λ = {130,100}, % /Lambda
8350   Ξ = {100,},    % /Xi
8351   Π = {100,},    % /Pi
8352   Σ = {100, 50}, % /Sigma
8353   Υ = {180,100}, % /Upsilon
8354   Φ = {130, 70}, % /Phi
8355   Ψ = {130, 50}, % /Psi
8356   Ω = { 50,},    % /Omega
8357   }
8358   </LatinModernRoman>
8359   <*CharisSIL>
8360   \SetProtrusion
8361   [ name      = Charis-default ]
8362   { encoding = {EU1,EU2,TU},
8363     family   = Charis SIL }
8364   {
8365   A = {50,50},
8366   Æ = {50,50},
8367   C = {50, },
8368   D = { ,50},
8369   F = { ,50},
8370   G = {50, },
8371   J = {100, },
8372   K = { ,50},
8373   L = { ,50},
8374   L̄ = { ,100},
8375   O = {50,50},
8376   Œ = {50, },
8377   P = { ,50},
8378   Q = {50,70},
8379   R = { ,50},
8380   Œ = { ,40}, % capital sharp s
8381   T = {50,50},
8382   V = {50,50},
8383   W = {50,50},
8384   X = {50,50},
8385   Y = {50,50},
8386   k = { ,50},
8387   l̄ = { ,150},
8388   r = { ,50},
8389   t = { ,50},
8390   v = {50,50},
8391   w = {50,50},
8392   x = {50,50},
8393   y = { ,50},
8394   1 = {150,150},
8395   2 = {50,50},
8396   3 = {50, },
8397   4 = {100,50},
8398   6 = {50, },
8399   7 = {50,80},
8400   9 = {50,50},
8401   . = { ,600},
8402   {,} = { ,500},
8403   : = { ,400},
8404   ; = { ,300},
8405   ! = { ,100},
8406   ? = { ,200},
8407   @ = {50,50},
8408   ~ = {200,250},
8409   \% = { ,50},
8410   * = {300,300},

```

8411 + = {200,250},  
 8412 / = { ,200},  
 8413 /backslash = {150,200},  
 8414 | = {200,200},  
 8415 - = {400,500}, % hyphen  
 8416 – = {200,300}, % endash  
 8417 — = {150,250}, % emdash  
 8418 ⎯ = {200,200}, % Horizontal Bar = \texttwelveudash  
 8419 - = {150,150}, % Figure Dash = \textthreequartersemdash  
 8420 \_ = {100,100},  
 8421 {=} = {100,100},  
 8422 ‘ = {300,400}, ’ = {300,400},  
 8423 “ = {300,300}, ” = {300,300},  
 8424 , = {400,400}, „ = {300,300},  
 8425 ‹ = {400,300}, › = {300,400},  
 8426 ‹‹ = {200,200}, ›› = {150,300},  
 8427 ¡ = {100, }, ¿ = {100, },  
 8428 ( = {200, }, ) = { ,200},  
 8429 < = {200,150}, > = {100,200},  
 8430 [ = {100, }, ] = { ,100},  
 8431 /braceleft = {200, }, /braceright = { ,300},  
 8432 † = { 80, 80},  
 8433 ‡ = {100,100},  
 8434 • = {200,200},  
 8435 ° = {150,200},  
 8436 ™ = {150,150},  
 8437 ¢ = { 50, },  
 8438 £ = { 50, },  
 8439 † = {200,200},  
 8440 © = {100,100},  
 8441 ® = {100,100},  
 8442 º = {100,200},  
 8443 ° = {200,200},  
 8444 ¬ = {200, 50},  
 8445 μ = { ,100},  
 8446 ¶ = { ,100},  
 8447 · = {300,400},  
 8448 <sup>1</sup> = {200,300},  
 8449 <sup>2</sup> = {100,200},  
 8450 <sup>3</sup> = {100,200},  
 8451 € = {100, },  
 8452 ± = {150,200},  
 8453 × = {200,200},  
 8454 ÷ = {250,250},  
 8455 /minus = {200,200},  
 8456 − = {200,200},  
 8457 % Cyrillic  
 8458 Б = { ,50},  
 8459 Г = { ,130},  
 8460 Ж = {50,50},  
 8461 З = {30,50},  
 8462 Л = {50, },  
 8463 У = {50,50},  
 8464 Ф = {50,50},  
 8465 Ч = {100, },  
 8466 Ъ = { ,50},  
 8467 Ь = { ,50},  
 8468 Э = {50,50},  
 8469 Ю = { ,40},  
 8470 Я = {50, },  
 8471 В = {50,50},  
 8472 € = {50, },  
 8473 Ъ = {50,100},  
 8474 € = {50, },  
 8475 Ъ = {50,50},

```

8476   Ѓ = { ,50},
8477   Є = {50,50},
8478   Ѕ = {100,100},
8479   Ь = {50,50},
8480   Ї = { ,50},
8481   Љ = { ,50},
8482   Њ = {50,80},
8483   Ћ = { ,80},
8484   Ќ = {50,50},
8485   Ѝ = {50, },
8486   Ў = {50,40},
8487   а = { ,50},
8488   б = {50, },
8489   в = { ,50},
8490   г = { ,50},
8491   Є = { ,100},
8492   Ѕ = {50,50},
8493   Ь = { ,70},
8494   Ї = { ,50},
8495   Љ = {50, },
8496   Њ = {50,50},
8497   Ћ = {50,50},
8498   Ќ = {50, },
8499   Ѝ = { ,50},
8500   Ў = { ,50},
8501   а = { ,50},
8502   б = {50, },
8503   в = {50, },
8504   г = { ,50},
8505   Є = { ,50},
8506   Ѕ = {50,50},
8507   Ь = {50, },
8508   Ї = { ,50},
8509   Њ = {50,50},
8510   Ћ = { ,50},
8511   Ќ = { ,50},
8512   Ѝ = { ,100},
8513   Ў = {100,100},
8514   а = {50,50},
8515   б = {50,70},
8516   в = { ,70},
8517   г = {50,30},
8518   Є = { ,50},
8519   Ѕ = { ,50},
8520   %   Д П Ц Ш Щ Ъ Ы Ь Ѓ Ѡ ѡ Ѣ ѣ Ѥ ѥ Ѧ
8521   %   в д ж з и м н п ц ш ы ю ѣ е Ѧ ѧ Ѩ ѩ Ѫ ѫ Ѭ ѭ
8522   % Greek
8523   Δ = {50,50},
8524   Ψ = {50,50},
8525   γ = {70,70},
8526   λ = {40,70},
8527   π = {40,50},
8528   ρ = { ,50},
8529   σ = { ,50},
8530   χ = {50,50},
8531 }
8532
8533 \SetProtrusion
8534 [ name = Charis-it ]
8535 { encoding = {EU1,EU2,TU},
8536   family = Charis SIL,
8537   shape = {it,sl} }
8538 {
8539   C = {50, },
8540   G = {50, },

```

8541 J = {50, },  
8542 L = {50,50},  
8543 O = {50, },  
8544 Œ = {50, },  
8545 Q = {50, },  
8546 S = {50, },  
8547 \$ = {50, },  
8548 T = {70, },  
8549 o = {50,50},  
8550 p = { ,50},  
8551 q = {50, },  
8552 t = { ,50},  
8553 w = { ,50},  
8554 y = { ,50},  
8555 1 = {150,100},  
8556 3 = {50, },  
8557 4 = {100, },  
8558 6 = {50, },  
8559 7 = {100, },  
8560 . = { ,700},  
8561 {,} = { ,600},  
8562 : = { ,400},  
8563 ; = { ,400},  
8564 ? = { ,150},  
8565 & = { ,80},  
8566 \% = {50,50},  
8567 \* = {300,200},  
8568 + = {250,250},  
8569 @ = {80,50},  
8570 ~ = {150,150},  
8571 / = { ,150},  
8572 /backslash = {150,150},  
8573 - = {300,400}, % hyphen  
8574 - = {200,300}, % endash  
8575 — = {150,200}, % emdash  
8576 \_ = { ,100},  
8577 {=} = {200,200},  
8578 ± = {150,200},  
8579 × = {250,250},  
8580 ÷ = {250,250},  
8581 ° = {150,200},  
8582 · = {300,400},  
8583 ‘ = {400,200}, ’ = {400,200},  
8584 “ = {300,200}, ” = {400,200},  
8585 , = {200,500}, „ = {150,500},  
8586 ‹ = {300,400}, › = {200,500},  
8587 « = {200,300}, » = {150,400},  
8588 ( = {200, }, ) = { ,200},  
8589 < = {200,200}, > = {200,200},  
8590 /braceleft = {300, }, /braceright = { ,200},  
8591 % Cyrillic  
8592 Ж = {50,30},  
8593 Л = {50, },  
8594 У = {50,30},  
8595 Ф = {50, },  
8596 Ч = {100, },  
8597 Ъ = { ,50},  
8598 Ь = { ,50},  
8599 Э = {50,50},  
8600 Я = {50, },  
8601 В = {50,50},  
8602 Ъ = {50,50},  
8603 Ъ = {140,100},  
8604 Ъ = {70,50},  
8605 Ъ = {50,80},

```

8606   Ḥ = { ,80},
8607   Ŧ = {50,50},
8608   Γ = {50,50},
8609   Δ = {50,30},
8610   Μ = {50, },
8611   Φ = {50, },
8612   Ψ = {50, },
8613   Ϛ = { ,50},
8614   ϛ = { ,50},
8615   Ϝ = { ,50},
8616   ϝ = {50, },
8617   Ϟ = {50,50},
8618   ϟ = { ,50},
8619   Ϡ = {50,50},
8620   ϡ = { ,50},
8621   Ϣ = {140,100},
8622   ϣ = {70,50},
8623   Ϥ = {50,70},
8624   ϥ = { ,70},
8625   % Greek
8626   Γ = { ,130},
8627   Δ = {50,50},
8628   Ψ = {50,50},
8629   γ = {70,70},
8630   λ = {40,70},
8631   π = {40,50},
8632   ρ = { ,50},
8633   σ = { ,50},
8634   χ = {50,50},
8635   }

```

The small caps glyph names in Charis SIL have changed with version 5.0 of the font. We try to get the names right both with LuaTeX (where we can simply query the font version) and with XeTeX (where we check for glyph name).

```

8636
8637 % quick and dirty -- maybe we'll promote this to a
8638 % regular key some time
8639 \define@key{MT@pr@c}{command}{\csname #1\endcsname}
8640
8641 % glyph names have changed with version 5.0 of Charis SIL:
8642 % before: /a.SC, /b.SC, ...
8643 % after: /a.sc, /b.sc, ...
8644 \ifx\MT@lua\undefined
8645   \gdef\MT@get@CHARIS@SC{
8646     % test whether glyph "a.sc" exists
8647     \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
8648       \gdef\MT@CHARIS@SC{sc}%
8649     \else
8650       \gdef\MT@CHARIS@SC{SC}%
8651     \fi
8652   }
8653 \else
8654   \gdef\MT@get@CHARIS@SC{
8655     \gdef\MT@CHARIS@SC{\MT@lua{
8656       % check font version
8657 % -- why doesn't this work?:
8658 %   f = font.getfont(font.current());
8659 %   i = fontloader.info(f.filename);
8660 %   if (tonumber(i.version) < 5) then;
8661 %   if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
8662     tex.print("SC");
8663   else;
8664     tex.print("sc");
8665   end

```

```

8666   }}
8667   }
8668 \fi
8669
8670 \SetProtrusion
8671   [ name      = Charis-sc,
8672     load      = Charis-default,
8673     command   = {MT@get@CHARIS@SC} ]
8674   { encoding = {EU1,EU2,TU},
8675     family   = Charis SIL,
8676     shape    = {sc} }
8677   {
8678     % A = {100,100}, % etc., doesn't work with \textsc
8679     /a.\MT@CHARIS@SC = {100,100},
8680     /c.\MT@CHARIS@SC = {50, },
8681     /d.\MT@CHARIS@SC = { ,50},
8682     /f.\MT@CHARIS@SC = { ,50},
8683     /g.\MT@CHARIS@SC = {50, },
8684     /j.\MT@CHARIS@SC = {100, },
8685     /k.\MT@CHARIS@SC = { ,50},
8686     /l.\MT@CHARIS@SC = { ,50},
8687     /f.l.\MT@CHARIS@SC = { ,50},
8688     /o.\MT@CHARIS@SC = {50,50},
8689     /oe.\MT@CHARIS@SC = {50, },
8690     /q.\MT@CHARIS@SC = {50,70},
8691     /r.\MT@CHARIS@SC = { ,50},
8692     /t.\MT@CHARIS@SC = {50,100},
8693     /v.\MT@CHARIS@SC = {50,50},
8694     /w.\MT@CHARIS@SC = {50,50},
8695     /x.\MT@CHARIS@SC = {50,50},
8696     /y.\MT@CHARIS@SC = {50,50}
8697   }
8698 /CharisSIL
8699 *PalatinoLinotype
8700 \SetProtrusion
8701   [ name      = palatino-default ]
8702   { encoding = {EU1,EU2,TU},
8703     family   = {PalatinoLinotype} }
8704   {
8705     A = {50,50},
8706     D = { ,50},
8707     J = {50, },
8708     K = { ,50},
8709     L = { ,50},
8710     O = {25, },
8711     T = {50,50},
8712     V = {50,50},
8713     W = {50,50},
8714     X = {50,50},
8715     Y = {50,50},
8716     b = { ,25},
8717     d = {25,30},
8718     f = { ,50},
8719     g = { ,100},
8720     k = { ,50},
8721     p = { ,50},
8722     q = {50, },
8723     r = { ,50},
8724     t = { ,50}, ◆ = { ,50}, ◆ = { ,50},
8725     v = {75,50},
8726     w = {50,50},
8727     x = {50,50},
8728     y = {50,70},
8729     1 = {100,50},

```

8730 2 = {25,50},  
 8731 4 = {50, },  
 8732 6 = {50, },  
 8733 9 = {25, },  
 8734 Æ = {100, },  
 8735 Œ = {25, },  
 8736 . = { ,700}, .. = { ,350}, ... = { ,150},  
 8737 {,} = { ,500},  
 8738 := { ,500},  
 8739 ; = { ,500},  
 8740 != { ,100}, !! = { ,100},  
 8741 ? = { ,200}, ? = { ,200},  
 8742 @ = {50,50},  
 8743 ~ = {200,250},  
 8744 & = {50,100},  
 8745 \% = {100,100},  
 8746 \* = {200,200},  
 8747 + = {250,250},  
 8748 ( = {100, }, ) = { ,300},  
 8749 / = {200,300},  
 8750 - = {400,500},  
 8751 \textendash = {300,300}, \textemdash = {200,200},  
 8752 \textquoteleft = {500,700}, \textquoteright = {500,700},  
 8753 \textquotedblleft = {300,400}, \textquotedblright = {300,400},  
 8754 \textbackslash = {200,300},  
 8755 \quotesinglbase = {400,400}, \quotedblbase = {400,400},  
 8756 \guilsinglleft = {400,400}, \guilsinglright = {300,500},  
 8757 \guillemotleft = {300,300}, \guillemotright = {200,400},  
 8758 \textexclamdown = {100, }, \textquestiondown = {100, },  
 8759 \textbraceleft = {400,200}, \textbraceright = {200,400},  
 8760 \textless = {200,100}, \textgreater = {100,200},  
 8761 ≤ = {200,100}, ≥ = {100,200},  
 8762 \textminus = {300,300},  
 8763 \texttrademark = {200,200},  
 8764 \textcopyright = {200,200},  
 8765 \textregistered = {200,200},  
 8766 \textdegree = {300,300},  
 8767 ¡ = {450,500}, ¬ = {250,150},  
 8768 ◆ = {150,250},  
 8769 · = {850, 700},  
 8770 ¶ = {100,0},  
 8771 × = {150, 300},  
 8772 ª = {300,300}, º = {300,300},  
 8773 ⁰ = {200,400},  
 8774 ¹ = {400,350}, ² = {200,300}, ³ = {250,400},  
 8775 ⁴ = {250,350}, ⁵ = {200,300}, ⁶ = {250,400},  
 8776 ⁷ = {200,450}, ⁸ = {250,400}, ⁹ = {200,350},  
 8777 ⁰ = {200,400},  
 8778 ₁ = {400,250}, ₂ = {200,300}, ₃ = {250,400},  
 8779 ₄ = {250,350}, ₅ = {200,300}, ₆ = {250,400},  
 8780 ₇ = {200,450}, ₈ = {250,400}, ₉ = {200,350},  
 8781 ± = {150,100}, ÷ = {300,300},  
 8782 þ = { ,25},  
 8783 ₰ = {300,450}, ₱ = {300,450},  
 8784 ₲ = {300,450}, ₳ = {300,450},  
 8785 ₴ = {200,250}, ₵ = {200,250},  
 8786 π = {50, },  
 8787 f = { ,50},  
 8788 № = {100,150},  
 8789 \textservicemark = {100,200},  
 8790 - = {400,500}, - = {400,500}, - = {200,300},  
 8791 - = {205,305}, - = {200,300}, - = {50,150},  
 8792 • = {125,200},  
 8793 % /a.sc = {50,50},  
 8794 }

```

8795
8796 \SetProtrusion
8797   [ name      = palatino-it  ]
8798   { encoding  = {EU1,EU2,TU},
8799     family    = {PalatinoLinotype},
8800     shape     = {it,s1} }
8801   {
8802     A = {50,50},
8803     Æ = {50, },
8804     B = {50, },
8805     C = {50, },
8806     D = {50,50},
8807     E = {50, },
8808     F = {50, },
8809     G = {50, },
8810     H = {50, },
8811     K = {50, },
8812     L = {50, },
8813     O = {50, },
8814     Œ = {50, },
8815     P = {50, },
8816     Q = {50, },
8817     R = {50, },
8818     S = {50, },
8819     $ = {50, },
8820     T = {100, },
8821     U = {50, },
8822     V = {100,50},
8823     W = {50, },
8824     X = {50, },
8825     Y = {100,50},
8826     b = { ,50},
8827     c = {25, },
8828     g = {75, },
8829     i = {25, },
8830     m = { ,50},
8831     n = { ,50},
8832     p = { ,25},
8833     q = {25, },
8834     x = { ,50},
8835     1 = {100, },
8836     2 = {50, },
8837     4 = {50, },
8838     7 = {50, },
8839     . = { ,500},    .. = { ,350},    ... = { ,200},
8840     {,} = { ,500},
8841     := { ,300},
8842     ; = { ,300},
8843     ? = { ,300},    ʔ = { ,300},
8844     & = {50,50},
8845     \% = {100,100},
8846     * = {200,200},
8847     + = {150,200},
8848     @ = {50,50},
8849     ~ = {200,150},
8850     ( = {200, },    ) = { ,200},
8851     / = {100,200},
8852     - = {300,500},
8853     \textendash    = {300,300},    \textemdash    = {200,200},
8854     \textquoteleft = {700,400},    \textquoteright = {700,400},
8855     \textquotedblleft = {500,300},    \textquotedblright = {500,300},
8856     _ = {100,100},
8857     \textbackslash = {100,200},
8858     \quotesinglbase = {500,500},    \quotedblbase = {400,400},
8859     \guilsinglleft = {400,400},    \guilsinglright = {300,500},

```

```

8860 \guillemotleft = {300,300}, \guillemotright = {300,300},
8861 \textexclamdown = {100, }, \textquestiondown = {200, },
8862 \textbraceleft = {200,100}, \textbraceright = {200,200},
8863 \textless = {300,100}, \textgreater = {200,100},
8864 ≤ = {200,100}, ≥ = {100,200},
8865 † = {450,500}, ‡ = {250,150},
8866 · = {850,700},
8867 ¶ = {100,0},
8868 × = {150,300},
8869 ª = {300,250}, º = {300,300}, ° = {300,250},
8870 º = {300,200},
8871 ¹ = {300,150}, ² = {350,200}, ³ = {250,150},
8872 ⁴ = {350,100}, ⁵ = {300,50}, ⁶ = {400,100},
8873 ⁷ = {400,50}, ⁸ = {250,50}, ⁹ = {300,50},
8874 ₀ = {300,300},
8875 ₁ = {300,350}, ₂ = {300,150}, ₃ = {250,250},
8876 ₄ = {400,200}, ₅ = {300,100}, ₆ = {450,200},
8877 ₇ = {450,150}, ₈ = {400,250}, ₉ = {400,200},
8878 ± = {150,100}, ÷ = {300,300},
8879 þ = {50, },
8880 † = {250,200}, ‡ = {250,200},
8881 ˙ = {300,450}, ˘ = {300,450},
8882 ˙ = {300,450}, ˘ = {300,450},
8883 - = {300,500}, - = {300,500}, - = {100,300},
8884 - = {125,305}, - = {200,300}, - = {125,150},
8885 • = {125,200}
8886 }
8887
8888 \SetProtrusion
8889 [ name = palatino-sc,
8890 load = palatino-default ]
8891 { encoding = {EU1,EU2,TU},
8892 family = {PalatinoLinotype},
8893 shape = sc }
8894 {
8895 a = {50,50},
8896 æ = {50, },
8897 b = {0,0},
8898 d = {0,0},
8899 f = {0,0},
8900 g = {0,0},
8901 j = {50, },
8902 l = { ,50},
8903 o = {0,0},
8904 p = {0,0},
8905 q = {0, },
8906 r = { ,0},
8907 t = {50,50},
8908 y = {50,50},
8909 fl = {0,50},
8910 ffl = {0,50},
8911 ◊ = {0,50},
8912 ◊ = {0,50}
8913 }
8914 /PalatinoLinotype
8915

```

## 17 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```

8916 (*test)
8917 \documentclass{article}
8918
8919 %% Here you can specify the font you want to test, using
8920 %% the commands \fontfamily, \fontseries and \fontshape.
8921 %% Make sure to end all lines with a comment character!
8922 \newcommand*\TestFont{%
8923   \fontfamily{ppl}%
8924   %% \fontseries{b}%
8925   %% \fontshape{it}% sc, sl
8926 }
8927
8928 \usepackage{ifthen}
8929 \usepackage[T1]{fontenc}
8930 \usepackage[latin1]{inputenc}
8931 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
8932
8933 \pagestyle{empty}
8934 \setlength{\parindent}{0pt}
8935 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
8936 \newcommand*\testprotrusion[2][ ]{%
8937   \ifthenelse{\equal{#1}{r}}{\#2}%
8938   lorem ipsum dolor sit amet,
8939   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
8940   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
8941   you know the rest%
8942   \ifthenelse{\equal{#1}{l}}{\#2}%
8943   \linebreak
8944   {\fontencoding{\encodingdefault}%
8945   \fontseries{\seriesdefault}%
8946   \fontshape{\shapedefault}%
8947   \selectfont
8948   Here is the beginning of a line, \dotfill and here is its end}\linebreak
8949 }
8950 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
8951 \def\stripprefix#1>{}
8952 \newcount\charcount
8953 \begin{document}
8954
8955 \microtypesetup{expansion=false}
8956
8957 {\centering The font in this document is called by:\\
8958 \texttt{\showTestFont}\par}\bigskip
8959
8960 \TestFont\selectfont
8961 This line intentionally left empty\linebreak
8962 %% A -- Z
8963 \charcount=65
8964 \loop
8965   \testprotrusion{\char\charcount}
8966   \advance\charcount 1
8967   \ifnum\charcount < 91 \repeat
8968 %% a -- z
8969 \charcount=97
8970 \loop
8971   \testprotrusion{\char\charcount}
8972   \advance\charcount 1
8973   \ifnum\charcount < 123 \repeat
8974 %% 0 -- 9
8975 \charcount=48
8976 \loop

```

```
8977 \testprotrusion{\char\charcount}
8978 \advance\charcount 1
8979 \ifnum\charcount < 58 \repeat
8980 %%
8981 \testprotrusion[r]{,}
8982 \testprotrusion[r]{.}
8983 \testprotrusion[r]{;}
8984 \testprotrusion[r]{:}
8985 \testprotrusion[r]{?}
8986 \testprotrusion[r]{!}
8987 \testprotrusion[l]{\textexclamdown}
8988 \testprotrusion[l]{\textquestiondown}
8989 \testprotrusion[r]{\}}
8990 \testprotrusion[l]{\{ }
8991 \testprotrusion{/}
8992 \testprotrusion{\char~\}
8993 \testprotrusion{-}
8994 \testprotrusion{\textendash}
8995 \testprotrusion{\textemdash}
8996 \testprotrusion{\textquoteleft}
8997 \testprotrusion{\textquoteright}
8998 \testprotrusion{\textquotedblleft}
8999 \testprotrusion{\textquotedblright}
9000 \testprotrusion{\quotesinglbase}
9001 \testprotrusion{\quotedblbase}
9002 \testprotrusion{\guilsinglleft}
9003 \testprotrusion{\guilsinglright}
9004 \testprotrusion{\guillemotleft}
9005 \testprotrusion{\guillemotright}
9006
9007 \newpage
9008 The following displays the current font stretched by 5%,
9009 normal, and shrunk by 5%:
9010
9011 \bigskip
9012 \newlength{\MTln}
9013 \newcommand*\teststring
9014 {ABCDEFGHIJKLMNQRSTUvwxyz0123456789}
9015 \settowidth{\MTln}{\teststring}
9016 \microtypesetup{expansion=true}
9017
9018 \parbox{1.05\MTln}{\teststring\linebreak\}
9019 \parbox{0.95\MTln}{\teststring}\par\bigskip
9020 \parbox{0.95\MTln}{\teststring}
9021
9022 \end{document}
9023 /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}
```

9024 *(\*Logo)*

Here's how the logo on the title page was created.<sup>29</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>30</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`.

9025 `\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.

9026 `\input bbox.sty`

`\tempdim` Allocate some `dimen` registers.

9027 `\newdimen\tempdim`

`\fboxrulei` Frame width of the box as  $\TeX$  sees it.

9028 `\newdimen\fboxrulei`

9029 `\fboxrulei=0.1pt`

`\fboxruleii` Frame width of the bounding box.

9030 `\newdimen\fboxruleii`

9031 `\fboxruleii=0.1pt`

`\kernboxheight` Height of the box indicating the kern.

9032 `\newdimen\kernboxheight`

9033 `\kernboxheight=5pt`

`\scalettoem` An auxiliary macro. Return a dimension relative to the `em`-width of the font. Requires `e-TeX`.

9034 `\setcommand\scalettoem#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).

9035 `\fontinstcc`

9036 `\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.

9037 `\ifdim\fontdimen6\font = 0pt`

9038 `\typeout{***-Warning:-no-fontdimen-6-specified-***^J%}`

9039 `***-setting-it-to-\pdffontsize\font \ifnum\pdfversion < 130 pt\fi-***}`

9040 `\fontdimen6\font=\pdffontsize\font \ifnum\pdfversion < 130 pt\fi\relax`

9041 `\fi`

9042 `\installfonts`

29 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.

30 Message ID: 42aa3687\$0\$24366\$9b4e6d93@newsread2.arcor-online.net

```

9043   \input_metrics{}{\logofont,\metrics\printbbs{#1}\relax}
9044   \endinstallfonts
9045 }
9046 \normalcc
      Layers.
9047 \makeatletter
9048 \def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
9049 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9050 \ifx\mt@order\undefined\let\mt@order\@empty\fi
9051 \xdef\mt@order{\mt@order[(Logo)]}
9052 \let\mtl@resources\@empty
9053 \def\mtl@register#1{%
9054   \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
9055   \expandafter\xdef\csname mtl@#1\endcsname{\the\pdfastobj\space 0 R }
9056   \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
9057   \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
9058   \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
9059 \mtl@register{canvas}
9060 \mtl@register{characters}
9061 \mtl@register{bounding-boxes}
9062 \mtl@register{TeX-boxes}
9063 \xdef\mt@order{\mt@order}
9064 \global\let\mtl@objects\mt@objects
9065 \def\togglelayer#1#2{%
9066   \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
9067   user{/Subtype/Link
9068     /BS << /Type/Border/W 0 >> /H/0
9069     /A << /S/SetOCGState
9070     /State[/Toggle \csname mtl@#1\endcsname] >>
9071   }#2\pdfendlink
9072 }

```

\printbbs Preparation.

```

9073 \setcommand\printbbs#1{%
9074   \setbox0\hbox{#1}%
9075   \leavevmode
9076   \kern-\fboxrulei
      The canvas in the natural width of the text minus protrusion, in color bgcolor.
9077   \mtl@layer{canvas}{%
9078     \getboundarychars#1\relax
9079     \tempdim=\dimexpr\wd0 - (\scaletom{\lpcode\font\firstchar}+
9080       \scaletom{\rpcode\font\lastchar})\relax
9081     \kern\dimexpr\scaletom{\lpcode\font\firstchar}\relax
9082     \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
9083       \hrule width \tempdim
9084         height \dimexpr\dp0+\ht0+0.15em\relax}%
9085     \kern-\tempdim
      The baseline, in color blcolor.
9086     \vbox{\color{blcolor}%
9087       \hrule width \tempdim
9088         height \fboxrulei}%
9089     }%
9090     \kern-\dimexpr\wd0 -\scaletom{\rpcode\font\lastchar}\relax

```

The string.

```

9091 \printbbs #1\relax\relax
9092 }

```

\getboundarychars Get first ....

```

9093 \def\getboundarychars#1#2\relax{%
9094   \def\firstchar{~#1}%
9095   \getlastchar#1#2\relax
9096 }

```

\getlastchar ... and last character.

```

9097 \def\getlastchar#1#2{%
9098   \ifx\relax#2\relax
9099     \def\lastchar{`#1}%
9100   \else
9101     \expandafter\getlastchar
9102   \fi #2%
9103 }

\printbss   Loop over all characters of the string.
9104 \def\printbss#1#2#3\relax{%
9105   \ifx\relax#1\relax
9106     \else
9107       \ifx\relax#2\relax
9108         \printbb{#1}{}%
9109       \else
9110         \printbb{#1}{#2}%
9111       \fi
9112     \expandafter\printbss
9113   \fi #2#3\relax
9114 }

\printbb   Record the kern between the current and the following character, then print the character. \kerning is a fontinst
           command.
9115 \setcommand\printbb#1#2{%
9116   \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
9117   \showboxes{#1}%
           This could be another application.
9118 %   \quad
9119 %   w: \the\scaletoe{\width{#1}},
9120 %   bb: \the\scaletoe{\bbleft{#1}}/%
9121 %   \the\scaletoe{\bbright{#1}},
9122 %   \the\scaletoe{\number\numexpr\width{#1}-\bbright{#1}\relax}
9123 %   h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
9124 }

\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' ≠ ',').
9125 \setcommand\showboxes#1{%
9126   \leavevmode
9127   \color{texcolor}%
           We have to record the width of the glyph.
9128   \setbox0\hbox{{\color{textcolor}#1}}%
9129   \global\tempdim=\wd0\relax
9130   \kern-\fboxrulei
           1. The  $\TeX$  box: Print a frame in color texcolor. This frame shows the glyph as  $\TeX$  sees it.
9131   \mtl@layer{TeX-boxes}{%
9132     \hbox{%
9133       \lower\dimexpr \dp0 + \fboxrulei\relax
9134       \hbox{%
9135         \vbox{%
9136           \hrule height\fboxrulei
9137           \hbox{%
9138             \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
9139             \phantom{\unhcopy0}%
9140             \vrule width\fboxrulei
9141           }%
9142           \hrule height\fboxrulei}}}%
9143     }%
           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.
9144   \kern-\wd0
9145   \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.
9146   \kern\dimexpr\scaletoe{\bbleft{#1}}-\tempdim-\fboxrulei\relax

```

3. *The bounding box*: will be printed in color `bbcolor`.

```

9147 \mtl@layer{bounding-boxes}{%
9148   {\color{bbcolor}%
9149   \hbox{%
9150     \lower\dimexpr-\scaletoe{\bbottom{#1}}+\fboxrulei\relax
9151     \hbox{%
9152       \vbox{%
9153         \hrule height\fboxrulei
9154         \hbox to \dimexpr\scaletoe{\numexpr
9155           \bbright{#1}-\bbleft{#1}\relax}+2\fboxrulei\relax{%
9156           \vrule height \dimexpr\scaletoe{\numexpr
9157             \bbtop{#1}-\bbottom{#1}\relax}%
9158             width\fboxrulei
9159           \hfill
9160           \vrule width\fboxrulei}%
9161         \hrule height\fboxrulei}}}%
9162     }%
9163     \kern-\dimexpr\fboxrulei+\fboxrulei\relax
9164   }%

```

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.

```

9165 \kern\scaletoe{\numexpr\width{#1}-\bbright{#1}\relax}%
9166 \mtl@layer{TeX-boxes}{%
9167   {\ifnum\thekern<0
9168     \color{kerncolor}%
9169     \kern\scaletoe{\thekern}%
9170     \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoe{\thekern}\relax
9171       height \kernboxheight}%
9172     \kern\scaletoe{\thekern}%
9173   \else
9174     \color{texcolor}%
9175     \ifnum\thekern=0 \else
9176       \lower\kernboxheight
9177       \hbox{%
9178         \vbox{%
9179           \hrule height\fboxrulei
9180           \hbox{%
9181             \vrule height \kernboxheight width\fboxrulei
9182             \kern\dimexpr\scaletoe{\thekern}-2\fboxrulei\relax
9183             \vrule width\fboxrulei
9184           }%
9185           \hrule height\fboxrulei}}}%
9186         \fi
9187       \fi
9188     }%
9189   }%
9190   % \kern-\fboxrulei
9191 }

```

```

9192 \newbox\logobox
9193 \def\printlogo{%
9194   \setbox\logobox=\hbox{\vbox{%
9195     \MakePercentComment

```

This is the Kepler MM font used in the logo.

```

9196   \def\logofont{pkpri9e10}
9197   \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
9198   \font\thelogofont=\logofont\space at 82pt

```

This would load the italic Palatino font instead.

```

9199 %\def\logofont{pplri}
9200 %\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
9201 %\edef\logofont{\logofont8r}
9202 %\font\thelogofont=\logofont\space at 78pt

```

Load the font.

```

9203   \thelogo font
      Protusion values (overdone for didactic reasons).
9204   \lcode\font`M=96
9205   \rcode\font`e=46

      Now we can generate the logo.
9206   \pdfliteral direct{/SXS gs}%
9207   \showlogo{Microtype}%
9208 %   \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
9209 %   \kern5pt\[\[3\baselineskip]
9210 %   \long\def\@makefnmark##1{%
9211 %     \leftskip 0pt
9212 %     \parindent 0pt
9213 %     \everypar{\parindent 0pt}%
9214 %     \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
9215 %   \footnotetext[1]{This graphic display on a
9216 %     \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
9217 %     their \togglelayer{bounding-boxes}{bounding boxes}
9218 %     and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
9219 %   }%
9220   \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
9221   \immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}%
9222   \immediate\pdfxform
9223     attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
9224     resources {/Properties <<\mtl@resources>>
9225               /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
9226     \logobox
9227 %   \vskip-2.5\baselineskip
9228 %   \leavevmode
9229 %   \togglelayer{characters}{%
9230 %     \pdfrefxform\pdflastxform
9231 %   }%
9232   \pdfannot\logodimens{%
9233     /Subtype/Widget /FT/Btn /T(Logo)
9234     %/F 4 % why did I say this?
9235     /AP << /N \the\pdflastxform\space 0 R >>
9236     /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9237         /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
9238         /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
9239         /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
9240     >> }%
9241   \vspace{3\baselineskip}
9242 }

      Our font.
9243 \pdfmapline{+pkpmrri8r10 Kep1MM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmrri8a10.pfb}
      Define colours (thered and thegreen are copied from microtype.dtx).
9244 \def\mtdefinecolors{
9245   \definecolor{thered}{rgb}{0.65,0.04,0.07}
9246   \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
9247   \colorlet{texcolor}{thegreen!50} % TeX boxes
9248   \colorlet{kerncolor}{texcolor} % negative kerns
9249   \colorlet{bbcolor}{thered!50} % bounding box
9250   \colorlet{bgcolor}{black!8} % canvas
9251   \colorlet{blcolor}{black!50} % baseline
9252   \colorlet{textcolor}{black!40} % text
9253 }

      Use with microtype.dtx
9254 \ifx\documentclass\@twoclasseserror
9255   \usepackage{xcdraw}{xcolor}
9256   \mtdefinecolors
9257 \else

```

## A.2 Document

Now we can start the document.

```

9258 \documentclass[10pt,a4paper]{ltxdoc}
9259 \providecommand\MakePercentComment{\relax}
9260 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
    Re-use the preamble from microtype.dtx.
9261 \usepackage{microtype-doc}
9262 \usepackage{attachfile}
9263 \makeatletter
9264 \pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
9265 \makeatother
9266 \begin{document}
    You are currently reading this.
9267 \DocInput{microtype-logo.dtx}
9268 \newpage
9269 And here it is:
9270 \vfill
9271 \begin{center}
9272 \printlogo \null
9273 \end{center}
9274 \vfill
9275 \expandafter\enddocument
9276 \fi
    That's it.
9277 /Logo

```

## B The letterspacing illustration

This is `microtype-1ssample.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
  - `\1ssample`: 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}
```

```

9278 \ifx\1ssample\undefined
9279 *1ssample

```

Upon popular request, here's how I've created the letterspacing illustration.<sup>31</sup>

### B.1 Macros

Rule width and image height and depth.

```

9280 \makeatletter
9281 \newdimen\1samount
9282 \newdimen\1srule
9283 \1srule=0.2pt
9284 \def\1sheight{8pt}
9285 \def\1sdepth{12pt}

```

<sup>31</sup> Note that the `1ssample` module will not be created when installing `microtype`. Instead, the source file `microtype-1ssample.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).

```

9286 \def\lsfont{\fontfamily{paca}\selectfont}
      Loop over all letters in <#2>, letterspacing them by <#1>.
9287 \def\dols#1#2{\lssamount=#1\relax \dolss#2\enddols}
9288 \def\dolss#1#2\enddols{%
9289   \ifx\empty#2\empty\divide\lssamount 2\fi
9290   \ls{#1}%
9291   \ifx\empty#2\empty\else \dolss#2\enddols \fi
9292 }

      One tikz picture for each letter.
9293 \def\ls#1{%
9294   \begin{tikzpicture}[remember picture,line width=\lssrule]
9295     \tikzstyle{every node}=[inner sep=0pt]

      The bounding box.
9296     \mts@layer{stuff}{%
9297       \node[draw=thegrey,
9298         fill=theshade,
9299         outer sep=\lssrule,
9300         anchor=base,
9301         font=\lsfont]{\phantom{#1}};
9302     }

      The letter.
9303     \node[anchor=base,font=\lsfont](#1){#1};

      Two auxiliary coordinates.
9304     \path (#1.south west) ++(+.5\lssrule,-.5\lssrule) coordinate (#1L);
9305     \path (#1.base east) ++(-.5\lssrule,-\lssdepth) coordinate (#1R);
9306     \mts@layer{stuff}{%

      Now draw the normal character width,
9307     \draw[color=thered!75,
9308       fill=thered!30,
9309       outer sep=\lssrule]
9310       (#1L) rectangle (#1R);
9311     \ifdim\lssamount>0pt
9312     \path (#1.base east) ++(+.5\lssamount,-6pt) coordinate (#1_1s);
9313     \path (#1R) ++(\lssamount+\lssrule,\lssdepth) coordinate (#1E);

      and the letter space.
9314     \draw[color=thered,
9315       fill=thered!50,
9316       outer sep=\lssrule]
9317       (#1R) ++(+\lssrule,+0pt) rectangle (#1E);
9318     \fi
9319   }
9320 \end{tikzpicture}%
9321 \ignorespaces
9322 }

      Draw the interword space.
9323 \def\lssp#1#2#3#4{%
9324   \begin{tikzpicture}[remember picture,line width=\lssrule,inner sep=0pt]
9325     \mts@layer{stuff}{%
9326       \tikzstyle{every draw}=[anchor=bottom]
9327       \coordinate(#1space) at (#2/2,\lssdepth/2);
9328       \coordinate(#1stretch) at (#2+#3/2,+0pt);
9329       \coordinate(#1shrink) at (#2-#4/2,+0pt);
9330       \draw[color=thegreen,fill=thegreen!50,use as bounding box]
9331         (0,0) rectangle ++(#2,\lssdepth);
9332       \draw[color=thegreen,fill=thegreen!30]
9333         (+#2,-\lssrule) rectangle ++(+#3,-4pt+\lssrule);
9334       \draw[color=thegreen,fill=thegreen!50]
9335         (+#2,-\lssrule) rectangle ++(-#4,-4pt+\lssrule);
9336       \draw[->,line width=0.3pt,shorten <=0.5\lssrule,color=thegreen!50]

```

```

9337         (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
9338     \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
9339         (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
9340     }%
9341 \end{tikzpicture}%
9342 \ignorespaces
9343 }

Layers.
9344 \def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
9345 \def\mts@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
9346 \ifx\mt@objects\undefined\let\mt@objects\@empty\fi
9347 \ifx\mt@order \undefined\let\mt@order \@empty\fi
9348 \xdef\mt@order{\mt@order[(Sheep)]}
9349 \let\mts@resources\@empty
9350 \def\mts@register#1{%
9351     \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
9352     \expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj\space 0 R }
9353     \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
9354     \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
9355     \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
9356 \mts@register{stuff}
9357 \mts@register{tracking}
9358 \mts@register{ispace}
9359 \mts@register{ospace}
9360 \mts@register{istretch}
9361 \mts@register{ishrink}
9362 \mts@register{ostretch}
9363 \mts@register{oshrink}
9364 \mts@register{okern}
9365 \mts@register{ligature}
9366 \mts@register{_compatibility}
9367 \xdef\mt@order{\mt@order]}

Anchor point for the arrow in the code.
9368 \newcommand\anchorarrow[1]{%
9369     \tikz[remember picture,overlay]\node(#1_c){};}

Add an arrow from code to image.
9370 \newcommand\add@arrow[5][left]{%
9371     \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
9372         \mts@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}%
9373     }

Toggle layer.
9374 \def\toggle@layer#1#2#3{%
9375     \pdfstartlink
9376     user{/Subtype/Link
9377         /BS << /Type/Border/W 0 >> /H/0
9378 %     /BS << /Type/Border/W 1 /S/D /D[4 1] >>
9379 %     /C[0.7 0.7 0.7] /H/0
9380     /Contents(Click to Toggle!)
9381     /A << /S/SetOCGState
9382         /State[/Toggle \csname mts@#1\endcsname] >> }%
9383     \rlap{#2}%
9384     {\fboxsep=0pt \fboxrule=0pt
9385     \mts@layer{stuff}{%
9386         \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
9387     \mts@layer{#1}{%
9388         \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}}%
9389     }%
9390     \pdfendlink
9391     }
9392 \newcommand\showarrow[2][ ]{%
9393     \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
9394     \toggle@layer{\@tempa}{\itshape #2}}

```

The environment for our illustration.

```

9395 \def\ls@sample#1{%
9396   \parskip 4pt \parindent 0pt
9397   \par
9398   \vskip4pt
9399   {\leftskip 15pt
9400    \mt@pseudo@margin{\color{theblue}Click on the image to show the kerns
9401     and spacings involved. Click on emphasised words in the text below
9402     to reveal the relation of image and code.\strut}
9403    \mt@layer{compatibility}%
9404     \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
9405      \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
9406     \mt@pseudo@margin{\color{thered}%
9407      If you had a \acronym{PDF} viewer that understands
9408      \acronym{PDF}\,{\smaller1.5}, you could hide the arrows selectively.}}
9409     \vskip-\mt@unvdimen}%
9410   \vskip-4pt
9411   \setlength\fbxsep{4pt}%
9412   \leavevmode
9413   \pdfstartlink
9414     user{/Subtype/Link
9415       /BS << /Type/Border/W 0 >> /H/0
9416       /A << /S/SetOCGState
9417         /State[/Toggle \mts@stuff] >> }%
9418     \fcolorbox{theframe}{theshade}%
9419     {\fontsize{34}{38}\selectfont #1}%
9420   \pdfendlink
9421   \par\medskip
9422   }%
9423   \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
9424 }

```

Now define the illustration to be used in the document.

```

9425 \def\lssample{%
9426   \ls@sample{%
9427     \dols{Opt}{Stop}
9428     \lssp{o}{0.45em}{0.25em}{0.15em}
9429     \dols{0.16em}{\stearing}\hskip-\dimexpr 0.08em+\lslrule\relax
9430     \lssp{i}{13.82pt}{4.65pt}{2.08pt}
9431     \dols{0.16em}{sheep}
9432     \dols{Opt}{!}
9433   }%

```

Don't forget to add the arrows.

```

9434   \vspace{-\baselineskip}
9435   \add@arrow{red}      {tracking}{lsamount_c.east}{a_ls}
9436   \add@arrow{red}      {okern}   {okernend_c.east}{p_ls}
9437   \add@arrow{green}    {ospace}  {ospace_c.east}  {ospace}
9438   \add@arrow{green}    {ispace}  {ispace_c.center}{ispace}
9439   \add@arrow{green!75} {istretch}{istretch_c.east}{istretch.north}
9440   \add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
9441   \add@arrow{green!75} {ostretch}{ostretch_c.east}{ostretch.north}
9442   \add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
9443   \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
9444 }
9445 \fi

```

This is for use with microtype.dtx

```

9446 \ifx\documentclass\@twoclasseserror
9447   \usepackage{tikz}
9448 \else

```

## B.2 Document

```

9449 \documentclass[10pt,a4paper]{ltxdoc}
9450 \expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}

```

```

Re-use the preamble from microtype.dtx.
9451 \usepackage{microtype-doc}
9452 \usepackage{attachfile}
9453 \usepackage{tikz}
9454 \makeatletter
9455 \pdfcatalog{/OCProperties << /OCGs [\mt@objects]
9456                               /D << /Order [\mt@order] /BaseState/OFF >> >> }
9457 \makeatother
9458 \begin{document}
  You are currently reading this.
9459 \DocInput{microtype-1ssample.dtx}
  Now show what we are able to do.
9460 \noindent
9461 Since a picture is worth a thousand words, probably even more if, in our
9462 case, it depicts a couple of letterspaced words, let's bring one to sum up
9463 these somewhat confusing options. Suppose you had the following settings
9464 (which I would in no way recommend; they are only for illustrative purposes):
9465 \begin{verbatim}
9466 \SetTracking
9467 [ no ligatures = {"\anchorarrow{nolig}"f},
9468   spacing      = {60"\anchorarrow{ispace}"0*,"%
9469                  -1"\anchorarrow{istretch}"00*," \anchorarrow{ishrink}"},
9470   outer spacing = {4"\anchorarrow{ospace}"50,"%
9471                  "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
9472   outer kerning = {"\anchorarrow{okernbegin}"*,"%
9473                  \anchorarrow{okernend}"*} ]
9474 { encoding = * }
9475 { 1"\anchorarrow{lsamount}"60 }
9476 \end{verbatim}
9477 and then write:
9478 \begin{verbatim}
9479 Stop \textls{stealing sheep}!
9480 \end{verbatim}
9481 this is the (typographically dubious) outcome:
9482
9483 \lssample
9484
9485 \noindent
9486 While the word 'Stop' is not letterspaced, the space between the letters in
9487 the other two words is expanded by the \showarrow[tracking]{tracking-amount}{red}
9488 of 160/1000\,em\,=\allowbreak\,0.16\,em.
9489 The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
9490 increased by 60\%, while its \showarrow[istretch]{stretch}{green} amount is
9491 decreased by 10\% and the \showarrow[ishrink]{shrink}{green} amount is left
9492 untouched.
9493 The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
9494 piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
9495 \showarrow[oshrink]{shrink}{green} by 0.15\,em.
9496 Note that there is no outer space after the text, since the exclamation mark
9497 immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
9498 of half the letterspace amount (0.08\,em) is added.
9499 Furthermore, one \showarrow[ligature]{grey} wasn't broken up, because we
9500 neglected to specify the '~|s|' in the |no ligatures| key.
9501
9502 \expandafter\enddocument
9503 \fi
9504 </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> ) . . . . .	86	<code>\MT@get@listname@</code> : don't check for empty attributes list . . . . .	88
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i> ) . . . . .	144	<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 . . . . .	151	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i> ) . . . . .	93
settings for Adobe Minion (contributed by <i>Harald Harders</i> ) . . . . .	152	<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 . . . . .	117	<code>\MT@permute</code> : don't use sets for empty encoding . . . . .	119
<code>\MT@declare@sets</code> : remove spaces around set name . . . . .	104	<code>\MT@setup@expansion</code> : issue an error instead of a warning, when pdfTeX version is too old for <code>autoexpand</code> . . . . .	134
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded . . . . .	86	<code>\MT@split@codes</code> : fix: allow zero and negative values . . . . .	63
<code>\MT@get@basefamily</code> : only remove suffix if it is 'x' or 'j' . . . . .	87	<code>\MT@use@set</code> : remove spaces around set name . . . . .	108

### 2004/10/03 **Version 1.2**

Font aliases: declare <code>cmor</code> as an alias of <code>cmr</code> . . . . .	142	<code>\MT@get@inh@list</code> : fix: set inheritance list <code>\globally</code> to <code>\@empty</code> . . . . .	90
Font sets: new: <code>allmath</code> and <code>basicmath</code> . . . . .	141	<code>\MT@get@listname@</code> : alternatively check for alias font name . . . . .	88
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding . . . . .	176	<code>\MT@get@size</code> : additional magic to catch some errors hijack <code>\set@fontsize</code> instead of <code>\setfontsize</code> . . . . .	106
add settings for Computer Modern Roman math symbols . . . . .	180	<code>\MT@loop</code> : fix: new macro, used instead of <code>\loop</code> . . . . .	49
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement . . . . .	59	<code>\MT@maybe@do</code> : also check for alias font name . . . . .	59
<code>\MT@get@basefamily</code> : also remove 'w' (swash capitals) . . . . .	87	<code>\MT@permute@@@@</code> : more sanity checks for <code>\SetProtrusion</code> and <code>\SetExpansion</code> . . . . .	120
<code>\MT@get@highlevel</code> : check whether defaults have changed . . . . .	104	<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 load option does no longer require to give a name, too . . . . .	114	<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> . . . . .	142	<code>\MT@load@list</code> : check whether list exists . . . . .	86

### 2004/11/12 **Version 1.4**

General: check for <code>pdfcpot</code> . . . . .	54	( <code>OT1</code> , <code>T1</code> , <code>lmr</code> ) . . . . .	157
don't use scratch registers in global definitions . . . . .	90	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . . .	128
use <code>\pickup@font</code> instead of <code>\define@newfont</code> as the hook for <code>\MT@setupfont</code> . . . . .	98	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too . . . . .	111
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> . . . . .	125	when reading files (reported by <i>Michael Hoppe</i> ) . . . . .	87
<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> ) . . .	127	form abczz (reported by <i>Georg Verweyen</i> ) . . . .	87
optimisation: use less <code>\expandafers</code> and <code>\csnames</code>	43	<code>\MT@get@slot</code> : don't define <code>\MT@char</code> globally (save stack problem) . . . . .	90
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl) . . . . .	151	<code>\MT@ifdimen</code> : don't set <code>\MT@count</code> globally (save stack problem) . . . . .	46
slanted like italics . . . . .	160	<code>\MT@setup@PDF</code> : new message if <code>\pdfoutput</code> is changed . . . . .	132
<code>\MT@checklist@family</code> : fix: don't try alias family name if encoding failed . . . . .	60	<code>\MT@use@set</code> : don't use undeclared font sets . . . .	108
<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> ) . . . . .	126	<code>\MT@get@highlevel</code> : don't test defaults if called after begin document . . . . .	104
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i> ) . . . . .	124	<code>\MT@scale@factor</code> : warning for factors outside limits	65
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
add note about <code>DVIoutput</code> option . . . . .	8	<code>\MT@set@ex@codes</code> : allow non-selected font expansion . . . . .	69
Inheritance: remove <code>\ss</code> from T1 list, add <code>\DJ</code> . . .	145	<code>\MT@set@pr@codes</code> : adjust protrusion factors before setting the inheriting characters . . . . .	62
Protrusion: settings for Bitstream Charter . . . . .	152	<code>\MT@setup@expansion</code> : defaults: calculate step as $\min(\text{stretch}, \text{shrink})/5$ . . . . .	133
<code>\DeclareMicrotypeAlias</code> : remove spaces around arguments . . . . .	109	defaults: turn off expansion for DVI output . . . .	133
<code>\MT@cfg@catcodes</code> : reset catcode of '=' (compatibility with Turkish babel) . . . . .	87	disable automatic expansion for DVI output . . .	134
<code>\MT@fix@catcode</code> : reset catcode of '^' (compatibility with chemsym) . . . . .	35		

2005/01/24 **Version 1.6**

General: defaults: turn off expansion for old pdfTeX versions . . . . .	127	tune CMR math letters (OML encoding) . . . . .	181
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 . . . . .	126	<code>\MT@get@inh@list</code> : correct message if selected is false . . . . .	90
restructure dtx file . . . . .	141	<code>\MT@set@ex@codes</code> : introduce factor option . . . .	69
test whether <code>\pickup@font</code> has changed . . . . .	100	<code>\MT@set@pr@codes</code> : introduce factor option . . . .	62
test whether numeric options receive a number	126	<code>\MT@setup@expansion</code> : disable automatic expansion for old pdfTeX versions . . . . .	134
use e-TeX's <code>\ifcsname</code> and <code>\ifdefined</code> if defined	44	<code>\MT@use@set</code> : retain current set if new set is undeclared . . . . .	108
Protrusion: add italic uppercase Greek letters . . .	160	<code>\MT@vinfo</code> : new macro instead of <code>\ifMT@verbose</code> . .	36
improve settings for numbers (pointed out by <i>Peter Muthesius</i> ) . . . . .	154		

2005/02/02 **Version 1.6a**

Documentation: add table of fonts with tailored protrusion settings . . . . .	21	reported by <i>Bernard Gaulle</i> ) . . . . .	90
<code>\MT@get@slot</code> : completely redone, hopefully more robust (compatible with frenchpro; problem		<code>\MT@pdftex@no</code> : new macro . . . . .	39
		<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older pdfTeX versions . . . . .	69

2005/03/23 **Version 1.7**

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i> ) . . . . .	105	Protrusion: fix: remove <code>\</code> from OT1, add <code>\textbackslash</code> to T1 encoding . . . . .	155
disallow automatic expansion if pdfTeX too old	117	<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i> ) . . . . .	109
fix: remove space after <code>autoexpand</code> . . . . .	117	<code>\Microtype@Hook</code> : new command for font package authors . . . . .	128
new value for verbose option: errors . . . . .	125	<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility . . . . .	128
shorter command names . . . . .	50	<code>\MT@begin@catcodes</code> : also use inside configuration commands . . . . .	87
warning when running in draft mode . . . . .	132		
Documentation: add hint about compatibility . . . .	26		
remove table of match order (now table 4 on page 88) . . . . .	12		

<code>\MT@cfg@catcodes</code> : reset catcode of ‘:’ (compatibility with french* packages) . . . . .	87	for composite character; no uncontrolled expansion . . . . .	96
<code>\MT@DeclareMicrotypeAlias</code> : may also be used inside configuration files . . . . .	109	<code>\MT@scale</code> : new macro: use e-TeX’s <code>\numexpr</code> if available . . . . .	50
<code>\MT@getListname@</code> : use <code>\@tfor</code> ( <i>Andreas Böhmann’s</i> idea) . . . . .	88	<code>\MT@set@ex@codes</code> : two versions of this macro . . . . .	69
<code>\MT@get@slot</code> : remove backslash hack . . . . .	90	<code>\MT@setup@expansion</code> : modify <code>\showhyphens</code> . . . . .	135
test for <code>\chardefed</code> commands . . . . .	91	<code>\MT@split@name</code> : don’t define <code>\MT@encoding</code> &c. globally . . . . .	59
test whether <code>\(encoding)\(…)</code> is defined . . . . .	91	<code>\MT@test@ast</code> : make it simpler . . . . .	104
<code>\MT@if@list@exists</code> : don’t define <code>\MT@#1@c@name</code> globally, here and elsewhere . . . . .	89	<code>\MT@try@order</code> : always check for size, too (suggested by <i>Andreas Böhmann</i> ) . . . . .	88
<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i> ) . . . . .	46	fix: also check for <code>//(series)/(shape)//</code> (reported by <i>Andreas Böhmann</i> ) . . . . .	88
<code>\MT@increment</code> : use e-TeX’s <code>\numexpr</code> if available . . . . .	50	<code>\MT@warn@code@too@large</code> : new macro: type out maximum protrusion factor . . . . .	65
<code>\MT@is@composite</code> : new macro: construct command		<code>\MT@warn@err</code> : new macro: for verbose=errors . . . . .	36

2005/06/23 **Version 1.8**

General: <code>\SetProtrusion</code> : new key: unit . . . . .	116	<code>\MT@find@file</code> : no longer wrap names in commands . . . . .	86
if font substitution has occurred, set up the substitute font, not the selected one . . . . .	98	<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters . . . . .	64
new option: config to load a different main configuration file . . . . .	127	<code>\MT@get@font@dimen@six</code> : new macro: test whether <code>\fontdimen 6</code> is defined . . . . .	62
new option: unit, by default character . . . . .	126	<code>\MT@get@listname@</code> : made recursive . . . . .	88
Documentation: add example for factor option . . . . .	13	<code>\MT@get@slot</code> : fix: expand active characters . . . . .	90
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i> ) . . . . .	15	test whether <code>\(encoding)\(…)</code> is defined made more robust . . . . .	91
add hint about error messages . . . . .	27	<code>\MT@get@unit</code> : new macro: get unit for codes . . . . .	66
Font aliases: declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code> . . . . .	142	<code>\MT@in@rlist</code> : made recursive . . . . .	48
Font sets: add U encoding to <code>allmath</code> . . . . .	141	<code>\MT@is@active</code> : new macro: translate inputenc-defined characters . . . . .	94
Inheritance: remove <code>\DJ</code> from T1 list (it’s the same as <code>\DH</code> ) . . . . .	145	<code>\MT@is@letter</code> : warning for non-ASCII characters . . . . .	93
Protrusion: add LY1 characters for Times . . . . .	160	<code>\MT@ledmac@setup</code> : character protrusion with <code>ledmac</code> . . . . .	52
settings for AMS math fonts . . . . .	184	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code> . . . . .	47
verified settings for slanted Computer Modern Roman . . . . .	169	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code> . . . . .	48
<code>\add@accent</code> : fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by <i>Stephan Hennig</i> ) . . . . .	100	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code> . . . . .	36
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font . . . . .	109	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30 . . . . .	39
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set . . . . .	108	<code>\MT@permute@00000</code> : add ranges to the beginning of the lists . . . . .	120
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining ASCII characters . . . . .	87	<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (reported by <i>Mark Rossi</i> ) . . . . .	50
<code>\MT@check@rlist</code> : made recursive . . . . .	121	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>hyperref</code> is loaded . . . . .	54
<code>\MT@curr@list@name</code> : new macro: current list type and name . . . . .	97	restore <code>csquotes</code> ’s active characters . . . . .	54
<code>\MT@declare@sets</code> : warning when redefining a set . . . . .	104	restore percent character if Spanish <code>babel</code> is loaded . . . . .	54
<code>\MT@define@set@key@</code> : use comma lists instead of token lists . . . . .	104	<code>\MT@split@codes</code> : get character width once only . . . . .	63
		<code>\MT@use@set</code> : fix: remove braces in first line . . . . .	108
		<code>\MT@xadd</code> : simplified . . . . .	47

2005/10/28 **Version 1.9**

General: <code>\DeclareMicrotypeSet</code> : new key: font . . . . .	106	option unit: rename value relative to character . . . . .	126
<code>\SetProtrusion</code> : value ‘relative’ renamed to ‘character’ for key unit . . . . .	116	Documentation: add hint about <code>verbatim</code> environment . . . . .	25
allow context-specific font setup . . . . .	98	add remark about Type 1 fonts required for automatic font expansion . . . . .	8
compatibility with TeX Live hack (reported by <i>Herbert Voß</i> ) . . . . .	38	Font aliases: declare <code>qpl</code> and <code>qtm</code> ( <code>qfonts</code> , TeX Gyre) as aliases of <code>ppl</code> resp. <code>ptm</code> . . . . .	142
disable microtype setup inside <code>hyperref</code> ’s <code>\pdfstringdef</code> (reported by <i>Hàn Thế Thành</i> ) . . . . .	55	Font sets: add OT4 encoding to text sets . . . . .	141
fix: use <code>true</code> as the default value . . . . .	123	add T5 encoding to text sets . . . . .	141

Inheritance: add list for OT4	146	<code>\MT@exp@two@n</code> : new macros: less <code>\expandafters</code>	44
add list for T5 (requested by <i>Hàn Thê Thành</i> )	147	<code>\MT@get@opt</code> : new key ‘preset’ to set all characters to the specified value before loading the lists	66
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	155	<code>\MT@is@active</code> : redone: use <code>\set@display@protect</code>	94
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	151	<code>\MT@is@letter</code> : using <code>\catcode</code> should be more efficient than inspecting the <code>\meaning</code>	93
settings for T5 encoded Computer Modern Roman	151	<code>\MT@maybe@do</code> : redone	59
<code>\DisableLigatures</code> : new command: disable ligatures (requires pdfTeX 1.30)	110	<code>\MT@rem@from@clist</code> : new macro: remove an item from a comma list	48
<code>\microtypecontext</code> : new command: change setup context in the document	101	<code>\MT@scale@factor</code> : generalised	65
<code>\MT@checklist@family</code> : fix: add two missing <code>\expandafters</code>	60	<code>\MT@setup@expansion</code> : disable expansion if both step and shrink are zero	134
<code>\MT@detokenize@c</code> : fix the $\TeX$ version	45	warning if user requested zero step	133
		<code>\MT@toks</code> : use instead of <code>\toks@</code>	41
		<code>\SetProtrusion</code> : (et al.) new key: font	111

2005/12/05 **Version 1.9a**

General: ‘ <i>file name</i> ’/‘ <i>line number</i> ’ as default list name	114	diately (requested by <i>Georg Verwey</i> )	104
new option: <code>deferssetup</code> , by default true	124	<code>\MT@get@highlevel</code> : no longer check whether defaults have changed	104
remove superfluous test whether <code>\pickup@font</code> has changed	100	<code>\MT@ifdefined@c@T</code> : new macros: true case only	44
Documentation: add explanation for error message in DVI mode	27	<code>\MT@ifint</code> : use <code>\pdfmatch</code> if available	45
add explanation for error message with non-Type 1 fonts	27	<code>\MT@ifstreq</code> : use <code>\pdfstrcmp</code> if available	46
Font aliases: declare <code>mbch</code> ( <code>mathdesign</code> ) as an alias of <code>bch</code>	143	<code>\MT@in@clist</code> : fix	48
Protrusion: fix: remove ‘_’ from OT1 encoding	156	<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	151	<code>\MT@is@feature</code> : new macro: check for pdfTeX feature	51
<code>\microtypesetup</code> : inside the preamble, accepts all package options	128	<code>\MT@map@clist@n</code> : following $\LaTeX$ 3	47
<code>\MT@check@font@cx</code> : optimise context-sensitive setup	101	<code>\MT@permute@#@#@#@#</code> : don’t define permutations for unused encodings	120
<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	51

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	55	<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	65

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	113
Protrusion: settings for URW Garamond	152	<code>\MT@define@code@key@size</code> : fix: embrace <code>\MT@tempsize</code> in <code>\csname</code> (bug introduced in v1.9b)	113

2006/05/05 **Version 1.9d**

Font sets: <code>md*</code> instead of <code>m</code> series in basic sets	141	tweak AMS settings	184
add QX encoding to text sets	141	<code>\DeclareCharacterInheritance</code> : fix: empty context	117
Inheritance: add list for QX encoding (contributed by <i>Maciej Eder</i> )	147	<code>\MT@detokenize@n</code> : new macro: use <code>\detokenize</code> if available	45
Protrusion: settings for QX encoding (contributed by <i>Maciej Eder</i> )	158	<code>\MT@get@ex@opt</code> : fix: evaluate preset	70
settings for Euro symbols (Adobe, ITC, marvosym)	192	<code>\MT@get@font@dimen</code> : warning for zero fontdimen	65
		<code>\MT@get@opt</code> : optimise: don’t reset when preset op-	

tion is set	66	\SetProtrusion: (et al.) optimise: unify keys for mandatory argument	111
set list name before presetting	66	(et al.) split keys of optional and mandatory argument	111
\MT@is@active: support for Unicode (inputenc/utf8)	94		
\MT@setupfont@hook: restore \% and \# when tex4ht is loaded (reported by <i>Peter Dyballa</i> )	54		

2006/07/28 **Version 1.9e**

General: fix: default value for activate: true	123	settings for Euler Roman font	188
Documentation: add hint about unknown encodings	26	\DeclareCharacterInheritance: new key 'inputenc' to set the input encoding	117
include LPL	244	\MT@rem@from@clist: model after \@removeelement	48
Font aliases: declare zeur and zeus (eulervm) as aliases of eur resp. eus (euler)	143	\MT@setup@: empty \MT@setup@ after use (compatibility with the combine class)	51
Inheritance: adapt to marvosym's changed encoding	149	\pickup@font: no tracing with trace package	99
Protrusion: complete settings for Euler Fraktur and Script fonts	191	\SetExpansion: new key: inputenc	111
fix: forgotten comma in mt-mvs.cfg; adapt to marvosym's changed encoding	192	\SetProtrusion: (et al.) new key: inputenc	111

2006/09/09 **Version 1.9f**

Protrusion: fix: euler-vm did not load euler settings	188	\MT@reset@context: only reset context if it has actually been changed	102
\MT@curr@list@name: fix: \MessageBreak must not be expanded	97	\MT@set@inh@list: fix: forgotten comma in the features list	118
\MT@gdef@n: new macros: global variants	44	\MT@set@named@keys: new macro: set name first, simplify parsing of optional argument	113
\MT@get@inh@list: fix: input encoding must be set after the inheritance list has been parsed	90	\SetProtrusion: (et al.) set catcodes before parsing optional argument	111
\MT@glet: new macro	43		

2007/01/14 **Version 2.0**

General: compatibility with listings: set catcode of backslash to zero (reported by <i>Steven Bath</i> )	55	new: smallcaps	141
compatibility with soul: register \textls and \lststyle	55	\DeclareMicrotypeBabelHook: new command: interaction with babel	110
new option: babel, by default false (language-dependent setup suggested by <i>Ulrich Dirr</i> )	124	\lststyle: fix: font switches don't pose a problem anymore	78
new option: letterspace, by default 100	126	fix: letterspacing commands may be nested	78
new package letterspace: a stripped-down version, containing the letterspacing commands only	1	new command: letterspacing	78
option 'babel': fix: switch off French babel's shorthands properly (reported by <i>Daniel Flipo</i> )	139	totally redone, using the new \letterspacefont	78
option 'babel': switch off Turkish babel's shorthands	139	\MT@declare@sets: fix: empty size list when redefining set	104
option 'unit', \SetProtrusion: deprecate value 'relative' completely	116	\MT@is@symbol: made even more robust	95
Documentation: add hint about how to increase font_max and font_mem_size	27	\MT@load@inputenc: sanitise catcodes before loading input encoding (problem with listings)	67
add hint about warning when tracking and expansion is applied to a font	28	\MT@pdf@tex@no: case 6: pdfTeX 1.40	40
add remark about 'draft' option disabling microtype (noted by <i>Michalis Miatidis</i> )	9	\MT@setup@noligatures: maybe disable \MT@noligatures after the preamble	138
qualify hint about web documents with regard to older pdfTeX versions	25	\MT@split@name: adjust to possible letterspacing	59
qualify hints about expansion error messages with regard to older pdfTeX versions	27	\SetExtraKerning: new command: additional kerning	112
Font sets: new: footnotesize and scriptsize	141	\SetExtraSpacing: new command: adjustment of interword spacing	112
		\SetTracking: new command: tracking	112
		\textls: new command: letterspacing	82
		starred version: remove spaces around text	82
		\tracingmicrotypeinpdf: 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	78
fix: letterspace package forgot to load <code>keyval</code>	41	<code>\MT@orig@pickupfont</code> : compatibility with CJK: also check for its definition	98
<code>\slig</code> : new command: protect ligatures in letter-spaced text	78	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	82

2007/07/14 **Version 2.2**

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i> )	51	<code>\MT@is@composite</code> : more robust: expand exactly once	96
new option: <code>copyfonts</code>	125	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	95
simplify key declarations	114	<code>\MT@ls@font</code> : use <code>\font@name</code> , not <code>\MT@font</code>	74
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	62
logo transparency and amusement	1	<code>\MT@set@inputenc@</code> : only load inputenc files if necessary	67
Font aliases: declare <code>chr</code> ( <code>chmath</code> ) as an alias of <code>bch</code> (reported by <i>Geoff Vallis</i> )	143	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	75
declare <code>fp9x</code> , <code>fp9j</code> (FPL Neu) as aliases of <code>ppl</code> [ <code>xj</code> ]	143	possibility to customise interword spacing	75
Font sets: default set for tracking: <code>smallcaps</code>	142	<code>\MT@setup@expansion</code> : warning if stretch or shrink aren't multiples of <code>step</code>	135
Inheritance: remove <code>'-'</code> → <code>'127'</code>	145	<code>\MT@setup@font</code> : don't call <code>\@enc@update</code> anymore	57
Protrusion: settings for Bitstream Letter Gothic	152	only add features that are available with the respective pdfTeX	57
Spacing: add sample	193	<code>\MT@setup@font@hook</code> : restore percent character if Galician <code>babel</code> is loaded	54
Tracking: add ligatures that are to be disabled	149	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	63
<code>\DeclareMicrotypeVariants</code> : new command	109	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	74
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	110	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	74
<code>\slig</code> : always defined	78	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	137
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	61	<code>\MT@with@babel@and@T</code> : also inspect class options	52
<code>\MT@copy@font</code> : optionally work on copies of fonts	57	<code>\pickup@font</code> : letterspace: setup inside group	99
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	87	<code>\SetTracking</code> : new key <code>'no ligatures'</code> to disable ligatures of letterspaced fonts	112
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	64	new keys <code>'spacing'</code> and <code>'outer spacing'</code> to adjust interword spacing (suggested by <i>Steven E. Harris</i> )	112
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	78	third argument may be empty	112
<code>\MT@ifdimen</code> : employ LuaTeX features if available	46	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	102
<code>\MT@ifint</code> : employ LuaTeX features if available	45		
<code>\MT@ifstreq</code> : employ LuaTeX features if available	46		
fix: $\LaTeX$ version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i> )	46		

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> )	101
fix: really switch off Turkish shorthands	139	<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code>	87
new value for verbose option: <code>silent</code> (suggested by <i>Karl Berry</i> )	126	<code>\MT@define@set@key@font</code> : font: single asterisk means normal font	106
turned some warnings into errors	125	<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	94
add letterspacing illustration	17	<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in v2.2)	84
<code>\do@subst@correction</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i> )	100	<code>\MT@plain</code> : letterspace: support for <code>eplain/miniltx</code>	38
<code>\slig</code> : redone: extract outer kerns from current letterspacing amount	78	<code>\MT@set@curr@os</code> : adjusting spaces made more reliable	76

<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> ) . . . . .	75	<code>\MT@SetTracking</code> : sanity check for value . . . . .	112
possibility to customise outer kerning (suggested by <i>Stephan Hennig</i> ) . . . . .	75	<code>\MT@setup@tracking</code> : enable protrusion when tracking is enabled . . . . .	136
		<code>\MT@tr@outer@l</code> : only change pre outer space if it contains <code>shrink</code> . . . . .	80

2008/02/29 **Version 2.3a**

General: fix test for <code>soul</code> under plain <code>T<sub>E</sub>X</code> . . . . .	56	too old for extensions . . . . .	137
Documentation: add hint about <code>babel</code> having to be loaded first . . . . .	26	<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 . . . . .	131
mention <code>soulutf8</code> . . . . .	30	<code>\MT@set@codes</code> : generalised . . . . .	63
Protrusion: adjust LMR quotation marks again . . . . .	157	<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>pdfT<sub>E</sub>X</code> is			

2008/06/04 **Version 2.3b**

<code>\MT@exp@gcs</code> : new macro: reduce save stack size . . . . .	43	also check for its definition . . . . .	98
<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 . . . . .	106	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i> ) . . . . .	75
<code>\MT@noligatures@</code> : fix: warning messages for unknown slots . . . . .	85	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode . . . . .	80
<code>\MT@orig@pickupfont</code> : compatibility with <code>CJKutf8</code> :		make <code>\spaceskip-aware</code> ( <code>ragged2e</code> ) . . . . .	80
		<code>\MT@tr@outer@r@</code> : additional test for horizontal mode . . . . .	81

2008/11/11 **Version 2.3c**

General: <code>LuaT<sub>E</sub>X</code> supported by default . . . . .	40	coding (reported by <i>Vasile Gaburici</i> ) . . . . .	147
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> ) . . . . .	81
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> ) . . . . .	79
move setup to the very end (for <i>Colin Rourke</i> ) . . . . .	139	<code>\MT@setup@expansion</code> : default step: 1 for <code>pdfT<sub>E</sub>X</code> versions $\geq 1.40$ . . . . .	133
<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> ) . . . . .	81
<code>\lstyle</code> : disable for <code>LuaT<sub>E</sub>X</code> . . . . .	78	fix: don't adjust in math mode (reported by <i>Christoph Bier</i> ) . . . . .	81
make invalid in math mode . . . . .	78	fix: don't adjust inside discretionary (reported by <i>Maverick Woo</i> ) . . . . .	81
<code>\microtypesetup</code> : select font after setup . . . . .	128	<code>\MT@tr@set@okern</code> : allow empty value for outer kerning . . . . .	83
<code>\MT@check@active@set</code> : warning for missing default sets . . . . .	128	<code>\textls</code> : make math mode aware . . . . .	82
<code>\MT@lua</code> : update for <code>LuaT<sub>E</sub>X 0.36</code> . . . . .	41		
<code>\MT@set@tr@codes</code> : allow zero tracking . . . . .	74		
<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	<i>Karl Karlsson</i> ) . . . . .	194
Expansion: settings for T2A encoding (contributed by <i>Karl Karlsson</i> ) . . . . .	150	<code>\MT@get@font@dimen@six</code> : fix: gobbling settings with tracking failed (reported by <i>Leo</i> ) . . . . .	62
Font sets: <code>sc*</code> instead of <code>sc</code> in <code>smallcaps</code> set . . . . .	141	<code>\MT@setup@</code> : make space-unaware (requested by <i>Marcin Borkowski</i> ) . . . . .	51
add T2A encoding . . . . .	141	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (first reported by <i>Christian Stark</i> ) . . . . .	53
Protrusion: settings for T2A encoding (contributed by <i>Karl Karlsson</i> ) . . . . .	158	<code>\MT@tr@outer@r@</code> : fix: set current kerning and spacing again (found by <i>Lars Rönnbäck</i> ) . . . . .	81
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*) . . . . . 158

2013/03/13 **Version 2.5**

- General: allow contexts for LuaTeX . . . . . 115
- disable ‘DVIoutput’ option for XeTeX . . . . . 124
- fix: check whether ‘*file*’/‘*line*’ list name already exists (reported by *Till A. Heilmann*) . . . . . 114
- 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 . . . . . 26
- add hint about spacing and `ragged2e` . . . . . 26
- add hint about `dtx` source code . . . . . 28
- include `microtype-logo.dtx` and `microtype-lssample.dtx` . . . . . 217
- Font aliases: declare `lmsy` and `lmm` as aliases of `cmsy` resp. `cmm` (reported by *Jonas Hogstrom*) . . . . . 142
- declare `zgmX` etc. (`garamondX`) as aliases of `ugm` . . . . . 143
- declare Latin Modern Roman (OpenType version) as alias of `lmr` when `fontspec` is loaded . . . . . 142
- declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) . . . . . 143
- Font sets: add EU1 and EU2 encodings . . . . . 141
- Inheritance: add rudimentary list for EU1 and EU2 . . . . . 148
- Protrusion: add default lists for EU1 and EU2 . . . . . 156
- improvements to Computer Modern Roman italics (contributed by *Hendrik Vogt*) . . . . . 161
- Tracking: add EU2 encoding to default list . . . . . 149
- `\DeclareCharacterInheritance`: allow more than one encoding . . . . . 117
- `\DeclareMicrotypeAlias`: ignore spaces . . . . . 109
- `\ifMT@nofamily`: info if settings are not family-specific (suggested by *Hàn Thê Thành*) . . . . . 61
- `\LoadMicrotypeFile`: remove all spaces in font name . . . . . 109
- `\sstyle`: fix: ensure to set up math fonts (reported by *RazorXsr*) . . . . . 78
- `\microtypecontext`: fix: ensure to set up math fonts (reported by *RazorXsr*) . . . . . 102
- `\MT@define@code@key@family`: compatibility with `fontspec`: remove its internal counter (reported by *Till A. Heilmann*) . . . . . 113
- `\MT@define@code@key@font`: scrub `fontspec` feature count (found by *Meho R*) . . . . . 113
- `\MT@do@font`: adapt for LuaTeX . . . . . 49
- adapt for XeTeX . . . . . 50
- `\MT@get@slot@:` adapt for LuaTeX (requested by *Georg Duffner*) . . . . . 91
- adapt for XeTeX . . . . . 91
- `\MT@if@outer@next`: fix: conflict with `amsmath` (reported by *Scott Pakin*) . . . . . 81
- `\MT@info@missing@char`: fix error message for XeTeX (reported by *Juan Acevedo*) . . . . . 65
- `\MT@is@charx`: compatibility with `xunicode` . . . . . 95
- `\MT@ledmac@setup`: fix to work with XeTeX (reported by *Maieul Rouquette*) . . . . . 52
- `\MT@ls@set@ls`: allow formulas in optional argument to `\textls` (fix by *Heiko Oberdiek*) . . . . . 83
- `\MT@register@subst@font`: only register if it isn’t registered already (reported by *George Gratzner* and *Josep Maria Font*) . . . . . 100
- `\MT@register@subst@font@cx`: only register if it isn’t registered already . . . . . 101
- `\MT@scrubfeatures`: compatibility with `fontspec`: remove its internal counter . . . . . 59
- `\MT@set@all@pr`: fix: remove space (found by *Meho R*) . . . . . 62
- `\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*) . . . . . 136
- `\MT@setupfont@hook`: restore `\%` and `\#` when `mathastext` is loaded (found by *Seamus Bradley*) . . . . . 54
- select font with `fontspec` (found by *Georg Duffner*) . . . . . 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 . . . . . 198
- `\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*) . . . . . 92
- `\MT@led@unhbox@line`: simplified . . . . . 52
- `\MT@ledmac@setup`: support for `eledmac` . . . . . 52
- `\MT@ls@outer@k`: add marker for tightly nested `letterspacing` . . . . . 84
- `\MT@set@tr@codes`: fix: load font for `fontspec` . . . . . 76
- `\MT@xspace`: fix outer spacing problem with `xspace` (reported by *Dave*) . . . . . 82

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*) . . . . . 54
- Documentation: add hint about partial incompatibility with `xeCJK` and `luatexja` . . . . . 27
- missing characters printed with `Charis SIL` . . . . . 198
- suggest to use `etoolbox` to patch `\verbatim` . . . . . 26
- Font sets: add TU encoding (notified by *Will*) . . . . . 27

<i>Robertson</i> .....	141	<code>\MT@is@xchar</code> : update for fontspec's TU encoding ..	95
add <code>si</code> and <code>scit</code> to <code>smallcaps</code> set (reported by <i>uli</i> )	141	<code>\MT@ledmac@setup</code> : support for <code>reledmac</code> .....	52
new: <code>allmath-nott</code> and <code>alltext-nott</code> (requested by <i>Karl Berry</i> ) .....	141	<code>\MT@luatex@no</code> : update for LuaTeX 0.85 (renamed primitives) .....	41
Inheritance: add TU encoding .....	148	<code>\MT@noligatures@</code> : use <code>luaotfload</code> function to keep/inhibit ligatures .....	85
Protrusion: add TU encoding to lists .....	156	<code>\MT@orig@pickupfont</code> : (in)compatibility with <code>luaotfload</code> : disable unknown slots warnings (reported by <i>Max</i> ) .....	98
Tracking: add TU encoding to default list .....	149	(in)compatibility with <code>xeCJK</code> : disable unknown slots warnings (reported by <i>HcN</i> ) .....	98
<code>\DeclareMicrotypeSet</code> : ignore spaces .....	103	compatibility with <code>xeCJK</code> : pretend that CJK wasn't loaded .....	99
<code>\DeclareMicrotypeSetDefault</code> : ignore spaces ...	108	<code>\MT@set@tr@codes</code> : use <code>luaotfload</code> 's <code>kernfactor</code> feature if available .....	75
<code>\DeclareMicrotypeVariants</code> : ignore spaces .....	109	<code>\MT@xspace</code> : fix outer spacing problem with (not only) algorithm (reported by <i>Henning</i> and <i>Ronnie Marksch</i> ) .....	82
<code>\lststyle</code> : fix: ensure to set up math fonts (reported by <i>kleenstar</i> ) .....	78	<code>\UseMicrotypeSet</code> : ignore spaces .....	107
<code>\microtypecontext</code> : allow activate shortcut (reported by <i>Karl Berry</i> ) .....	101		
<code>\MT@declare@sets</code> : fix: undefine lists for redefining	104		
<code>\MT@do@font</code> : speed up for LuaTeX .....	49		
<code>\MT@engine</code> : fix test with LuaTeX 0.85 .....	39		
<code>\MT@fontspec@setup</code> : fix <code>\MT@if@fontspec@font ABD</code>	53		
<code>\MT@get@slot@</code> : fix: could fail with $X_{\text{TeX}}$ (reported by <i>Christopher Schramm</i> ) .....	91		

2016/05/14 **Version 2.6a**

General: fixes for <code>letterspace</code> package with LuaTeX	49	<code>VofB</code> ) .....	49
<code>\MT@do@font</code> : fix lua function (reported by <i>Herbert</i> )		<code>\MT@ls@fontspec@font</code> : fix for value of $\pm 1000$ ...	77

2017/07/07 **Version 2.7**

General: drop <code>luatexbase</code> with recent $\LaTeX$ .....	43	<code>\MT@check@range@</code> : don't warn for override if conflicting list is loaded .....	121
warning with <code>minimal</code> class .....	51	<code>\MT@is@composite</code> : compatibility with $\LaTeX$ 2017/01/01 ( <code>\DeclareUnicodeComposite</code> ) (reported by <i>Ulrike Fischer</i> and <i>jcr</i> ) .....	96
Documentation: add remark about automatic font expansion with <code>dvilualatex</code> .....	8	<code>\MT@ls@fontspec@font</code> : fix for ' <code>file:font</code> ' spec (reported by <i>Reinhard Kotucha</i> ) .....	77
mention that additional kerning does not work in math mode (discovered by <i>'Daniel'</i> ) .....	18	<code>\MT@permute@@@@@</code> : don't warn for override if conflicting list is loaded .....	121
Font aliases: declare aliases for <code>newpx</code> .....	143	<code>\MT@reset@ef@codes</code> : only reset <code>\efcodes</code> for older LuaTeX versions .....	69
declare aliases for <code>newtx</code> .....	143	<code>\MT@setup@expansion</code> : don't disable automatic expansion for DVI output with LuaTeX .....	134
declare aliases for <code>tempora</code> .....	143	<code>\MT@tikz@setup</code> : compatibility with <code>tikz</code> (again) ..	53
declare aliases for <code>XCharter</code> .....	143	<code>\MT@warn@tracking@DVI</code> : don't warn for letterspacing in DVI mode with LuaTeX .....	137
declare Latin Modern Roman as alias of <code>lmr</code> with new $\LaTeX$ format (reported by <i>Ulrike Fischer</i> )	142		
Protrusion: automatically choose correct names for Charis SIL small caps (reported by <i>'lcomdata'</i> )	210		
<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> ) .....	78		

2018/01/14 **Version 2.7a**

General: disallow non-automatic expansion with LuaTeX .....	117	<code>\MT@get@highlevel</code> : test whether <code>\...default</code> is defined .....	104
<code>\MT@auto</code> : remove <code>'autoexpand'</code> for LuaTeX 1.0.6 (reported by <i>Ulrike Fischer</i> ) .....	134	<code>\MT@get@slot</code> : expand active characters earlier ...	90
with LuaTeX, font expansion is always automatic (as confirmed by <i>Hans Hagen</i> ) .....	134	<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> ) .....	94

## 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 . . . . .	<b>8</b>	letterspace . . . . .	<b>8</b>
	activate . . . . .	<b>6</b>	protrusion . . . . .	<b>6</b>
	auto . . . . .	<b>7</b>	selected . . . . .	<b>8</b>
	babel . . . . .	<b>9</b>	shrink . . . . .	<b>8</b>
	config . . . . .	<b>9</b>	spacing . . . . .	<b>6</b>
	draft . . . . .	<b>9</b>	step . . . . .	<b>8</b>
	expansion . . . . .	<b>6</b>	stretch . . . . .	<b>8</b>
	factor . . . . .	<b>7</b>	tracking . . . . .	<b>6</b>
	final . . . . .	<b>9</b>	unit . . . . .	<b>7</b>
	kerning . . . . .	<b>6</b>	verbose . . . . .	<b>9</b>
<b>Commands</b>	\DeclareCharacterInheritance . . . . .	<b>20</b>	\SetExtraSpacing . . . . .	<b>19</b>
	\DeclareMicrotypeAlias . . . . .	<b>21</b>	\SetProtrusion . . . . .	<b>13</b>
	\DeclareMicrotypeBabelHook . . . . .	<b>23</b>	\SetTracking . . . . .	<b>15</b>
	\DeclareMicrotypeSet* . . . . .	<b>10</b>	\UseMicrotypeSet . . . . .	<b>12</b>
	\DeclareMicrotypeSet . . . . .	<b>10</b>	\Tslig . . . . .	<b>24</b>
	\DeclareMicrotypeSetDefault . . . . .	<b>12</b>	\Tstyle . . . . .	<b>23</b>
	\DeclareMicrotypeVariants* . . . . .	<b>20</b>	\microtypecontext . . . . .	<b>22</b>
	\DeclareMicrotypeVariants . . . . .	<b>20</b>	\microtypesetup . . . . .	<b>9</b>
	\DisableLigatures . . . . .	<b>24</b>	\textls* . . . . .	<b>23</b>
	\LoadMicrotypeFile . . . . .	<b>22</b>	\textls . . . . .	<b>23</b>
	\SetExpansion . . . . .	<b>14</b>	\textmicrotypecontext . . . . .	<b>22</b>
	\SetExtraKerning . . . . .	<b>18</b>		
<b>A</b>	a0poster (package) . . . . .	<b>106</b>	algorithm (package) . . . . .	<b>235</b>
	activate (option) . . . . .	<b>6, 123, 231</b>	amsmath (package) . . . . .	<b>234</b>
	\add@accent . . . . .	<b>2907</b>	amssymb (package) . . . . .	<b>184</b>
	\adjustspacing . . . . .	<b>4277</b>	article (package) . . . . .	<b>22</b>
	ae (package) . . . . .	<b>21, 142</b>	auto (option) . . . . .	<b>7, 124</b>
<b>B</b>	babel (option) . . . . .	<b>9, 23, 26, 31, 124, 231</b>		<b>54, 87, 110, 138, 139, 144, 228, 229, 231–233</b>
	babel (package) . . . . .	<b>2, 3, 5, 17, 23, 24, 26, 53,</b>		
<b>C</b>	chemsym (package) . . . . .	<b>228</b>	config (option) . . . . .	<b>9, 20, 32, 127, 229</b>
	chmath (package) . . . . .	<b>21, 143, 232</b>	contour (package) . . . . .	<b>132</b>
	CJK (package) . . . . .	<b>27, 56, 98, 99, 232, 233, 235</b>	\copyfont . . . . .	<b>1013</b>
	CJKutf8 (package) . . . . .	<b>99, 233</b>	copyfonts (option) . . . . .	<b>57, 124, 125, 232</b>
	cm-super (package) . . . . .	<b>8</b>	crop (package) . . . . .	<b>132</b>
	color (package) . . . . .	<b>9, 132</b>	csquotes (package) . . . . .	<b>29, 53, 54, 155, 229</b>
	combine (package) . . . . .	<b>139, 231</b>	\curr@fontshape . . . . .	<b>2851, 2853, 2855, 2862, 2905</b>
<b>D</b>	\DeclareCharacterInheritance . . . . .	<b>20, 41, 3722</b>	deferssetup (option) . . . . .	<b>54, 124, 128, 230</b>
	\DeclareMicrotypeAlias . . . . .	<b>21, 43, 71, 3310</b>	\define@newfont . . . . .	<b>2844, 2852, 2856, 2866</b>
	\DeclareMicrotypeBabelHook . . . . .	<b>23, 45, 64, 3376</b>	\DisableLigatures . . . . .	<b>24, 40, 62, 3349</b>
	\DeclareMicrotypeSet . . . . .	<b>10, 32, 57, 59, 72, 3024</b>	\do@subst@correction . . . . .	<b>2904</b>
	\DeclareMicrotypeSet* . . . . .	<b>10, 3024</b>	docstrip (package) . . . . .	<b>34</b>
	\DeclareMicrotypeSetDefault . . . . .	<b>12, 34, 61, 3265</b>	draft (option) . . . . .	<b>9, 124, 125, 231</b>
	\DeclareMicrotypeVariants . . . . .	<b>20, 42, 58, 63, 3295</b>	dsfont (package) . . . . .	<b>62</b>
	\DeclareMicrotypeVariants* . . . . .	<b>20</b>	DVIoutput (option) . . . . .	<b>8, 9, 124, 228, 234</b>
<b>E</b>	eco (package) . . . . .	<b>21, 142</b>	euler (package) . . . . .	<b>97, 188, 231</b>
	\efcode . . . . .	<b>1535, 1561, 1562, 1602, 1604</b>	eulervm (package) . . . . .	<b>21, 143, 188, 231</b>
	eledmac (package) . . . . .	<b>52, 234</b>	euroitc (package) . . . . .	<b>192</b>
	eplain (package) . . . . .	<b>24, 31, 34, 38, 52, 131, 232</b>	europs (package) . . . . .	<b>192</b>
	e-TeX (engine) . . . . .	<b>32,</b>	eurosans (package) . . . . .	<b>192</b>
		<b>38, 40, 44, 45, 50, 64, 81, 217, 228–230, 232</b>	\expandglyphsinfont . . . . .	<b>1529</b>
	etoolbox (package) . . . . .	<b>26, 234</b>	expansion (option) . . . . .	<b>6, 11, 123, 133</b>
	euca1 (package) . . . . .	<b>189</b>	exscale (package) . . . . .	<b>235</b>
	eufrak (package) . . . . .	<b>191</b>		

- F** `\f@family` ..... 2858, 2859  
`\f@size` ..... 1902, 2851, 2853, 2855, 2862, 2905  
`factor` (option) ..... 7, 13, 32, 126, 228  
`fancyvrb` (package) ..... 26  
`final` (option) ..... 9, 33, 124, 125, 227  
`fix-cm` (package) ..... 59  
`\fmtversion` ..... 355  
`\font` ..... 971, 1804, 1949  
`\font@name` ..... 111, 1018, 1020, 1022,  
1024, 1034, 1035, 1794, 1802, 1806, 1810,  
1848, 1854, 1859, 1862, 1935, 1949, 1961,  
1963, 1968, 1970, 1973, 1976, 1979, 2014,  
2069, 2206, 2208, 2844, 2852, 2861, 2866,  
2886, 2888, 2890, 2906, 2919, 2920, 2937, 2942
- G** `garamondx` (package) ..... 21, 143, 234  
`german` (package) ..... 35, 227  
`\glb@currsiz` ..... 2968
- H** `hfoldsty` (package) ..... 21, 142
- I** `IEEEtran` (package) ..... 98  
`\iffontchar` ..... 1323, 2780, 2792, 2793  
`\ifMT@auto` ..... 297, 4470, 4527  
`\ifMT@babel` ..... 297, 4603, 4731  
`\ifMT@do` ..... 1067, 1134, 1770, 2253  
`\ifMT@document` ..... 338, 3833  
`\ifMT@draft` ..... 297, 4374  
`\ifMT@expansion` ..... 297, 4432, 4524, 4560  
`\ifMT@fontspec` .....  
. 829, 842, 853, 888, 1048, 3095, 3500, 3526  
`\ifMT@if` ..... 296,  
858, 865, 895, 927, 2933, 3940, 4751, 4754  
`\ifMT@inannot` ..... 99  
`\ifMT@inlist` . . . 614, 659, 756, 786, 989, 1101,  
1118, 1127, 1145, 1163, 1768, 2357, 2368,  
2417, 2466, 2920, 2927, 2939, 2994, 3006, 3332  
`\ifMT@kerning` ..... 297, 4617, 4744
- J** `jurabib` (package) ..... 98
- K** `kerning` (option) ..... 6, 11, 31, 124  
`keyval` (package) ..... 41, 63, 144, 232  
`\knaccode` ..... 1719, 1720, 1730, 1733, 1739
- L** `ledmac` (package) ..... 29, 32, 52, 98, 229  
`ledpar` (package) ..... 29  
`\leftmargin` ..... 799  
`letterspace` (option) 8, 17, 23, 24, 31, 45, 74, 126, 231  
`letterspace` (package) ..... 1,  
24, 31, 34, 36, 38, 53, 75, 76, 78, 231–233, 235  
`\letterspacefont` ..... 1810  
`lineno` (package) ..... 52  
`listings` (package) . . . . 26, 55, 67, 87, 230, 231, 234  
`lmodern` (package) ..... 8, 157  
`\LoadMicrotypeFile` ..... 22, 44, 74, 3328, 4379  
`\lpcode` ..... 1215, 1251, 1252, 1455, 1458, 1837
- M** `marvosym` (package) . . . 21, 31, 34, 149, 192, 230, 231  
`mathastext` (package) ..... 54, 234  
`mathdesign` (package) ..... 21, 143, 230  
`memoir` (package) ..... 22, 98  
`\Microtype@Hook` ..... 128, 4208  
`\microtypecontext`  
. . . 22, 47, 916, 2963, 2974, 4381, 4714, 4717  
`microtypecontext` (environment) ..... 22  
`\microtypesetup` ..... 9, 46, 4212, 4380  
`MiKTeX` (distribution) ..... 28  
`miniltx` (package) ..... 24, 31, 34, 38, 52, 232
- `\fontcharwd` ..... 1278, 1282, 1288  
`\fontdimen` ..... 1201,  
1208, 1313, 1318, 1417, 1420, 1823, 1824,  
1854, 1961, 2007, 2008, 2015, 2016, 2022,  
2026, 2033, 2043, 2047, 2050, 2069, 2219, 2222  
`fontinst` (package) ..... 195, 217, 219  
`fontinstallationguide` (package) ..... 174  
`\fontname` ..... 1802, 1806  
`fontspec` (package) ..... 6, 25, 27, 30, 43,  
49, 53, 59, 77, 95, 105, 113, 142, 148, 234, 235  
`fouri` (package) ..... 62  
`french` (package) ..... 229  
`frenchpro` (package) ..... 95, 228, 232
- `\glb@settings` ..... 1938  
`graphics` (package) ..... 9, 132
- `hyperref` (package) . . . . 9, 54, 55, 98, 132, 229, 232
- `\ifMT@nofamily` ..... 1173, 1180  
`\ifMT@noligatures` ..... 297, 4697  
`\ifMT@nonselected` ..... 1511, 2479, 2487  
`\ifMT@noreset` ..... 2548, 2565, 2578, 2592, 2653  
`\ifMT@opt@auto` ..... 3951, 4474, 4504  
`\ifMT@opt@DVI` ..... 3951, 4399  
`\ifMT@opt@expansion` ..... 3951, 4428, 4561  
`\ifMT@protrusion` ..... 297, 793, 816, 4411, 4576  
`\ifMT@selected` ..... 297, 4510, 4530  
`\ifMT@spacing` ..... 297, 4585, 4602  
`\ifMT@tracking` ..... 297, 4572, 4632  
`\ifMT@xunicode` ..... 829, 2737, 2784  
`ifpdf` (package) ..... 9, 99, 132  
`\iftracingmicrotypeinpdfall` ..... 105  
`\ignoreligaturesinfont` ..... 2257  
`inputenc` (package) .....  
. . 14, 26, 31, 55, 67, 90, 94, 97, 229, 231, 232
- `\nbcrcode` ..... 1712, 1713, 1729, 1732, 1738  
`\nbscode` ..... 1633, 1634, 1657, 1661, 1667
- `\lslig` ..... 24, 54, 1954, 1957  
`\lststyle` ..... 23, 51, 918, 943, 948,  
1936, 2145, 2151, 2156, 2159, 2169, 2173, 4382  
`luaotfload` (package) .....  
. . . . 27, 30, 43, 49, 53, 75, 85, 92, 234, 235  
`LuaTeX` (engine) ..... 1,  
4, 6–8, 12–16, 20, 23–31, 34, 37–41, 43, 45,  
49, 53, 54, 69, 74–76, 82, 84, 85, 91, 115,  
125, 130, 134, 137, 142, 144, 148, 156, 232–235  
`luatexbase` (package) ..... 43, 234, 235  
`luatexja` (package) ..... 27, 98, 234, 235  
`luatextra` (package) ..... 234
- `minimal` (package) ..... 51, 56, 235  
`MinionPro` (package) ..... 105  
`\MT@font` ..... 397, 992,  
1016, 1203, 1314, 1327, 1782, 2420, 2492, 2891  
`\MT@abbr@ex` ..... 732  
`\MT@abbr@ex@c` ..... 732  
`\MT@abbr@ex@inh` ..... 732  
`\MT@abbr@kn` ..... 732  
`\MT@abbr@kn@c` ..... 732  
`\MT@abbr@kn@inh` ..... 732  
`\MT@abbr@nl` ..... 732

- \MT@abbr@pr ..... [732](#)  
 \MT@abbr@pr@c ..... [732](#)  
 \MT@abbr@pr@inh ..... [732](#)  
 \MT@abbr@sp ..... [732](#)  
 \MT@abbr@sp@c ..... [732](#)  
 \MT@abbr@sp@inh ..... [732](#)  
 \MT@abbr@tr ..... [732](#)  
 \MT@abbr@tr@c ..... [732](#)  
 \MT@active@features .....  
     [1026](#), [2921](#), [2924](#), [2936](#), [2947](#), [2983](#), [2993](#),  
     [3357](#), [4412](#), [4525](#), [4573](#), [4586](#), [4618](#), [4706](#), [4707](#)  
 \MT@addto@annot ..... [99](#)  
 \MT@addto@setup ..... [777](#),  
     [778](#), [869](#), [1430](#), [2835](#), [2963](#), [2964](#), [3587](#),  
     [3605](#), [4065](#), [4213](#), [4373](#), [4663](#), [4705](#), [4730](#), [4763](#)  
 \MT@adjustspacing ..... [4273](#), [4274](#), [4526](#)  
 \MT@afteraftergroup ..... [1866](#), [1874](#), [1883](#)  
 \MT@auto ..... [1521](#), [4468](#), [4484](#), [4494](#), [4500](#)  
 \MT@auto@ ..... [1521](#), [1531](#), [1575](#), [4468](#)  
 \MT@autofalse ..... [299](#), [4482](#), [4493](#)  
 \MT@autotru ..... [299](#), [4167](#), [4170](#)  
 \MT@babelfalse ..... [306](#)  
 \MT@babeltrue ..... [306](#)  
 \MT@begin@catcodes ..... [2358](#), [2359](#), [2399](#),  
     [3025](#), [3235](#), [3266](#), [3297](#), [3311](#), [3336](#), [3352](#),  
     [3386](#), [3402](#), [3424](#), [3445](#), [3459](#), [3728](#), [4183](#), [4184](#)  
 \MT@cat ..... [1426](#), [2826](#), [2828](#)  
 \MT@cfg@catcodes ..... [929](#), [1448](#), [2384](#), [2401](#)  
 \MT@char ..... [1240](#),  
     [1241](#), [1251](#), [1252](#), [1257](#), [1258](#), [1261](#), [1263](#),  
     [1278](#), [1279](#), [1282](#), [1455](#)–[1459](#), [1561](#), [1562](#),  
     [1564](#), [1565](#), [1602](#)–[1604](#), [1633](#), [1634](#), [1640](#),  
     [1641](#), [1647](#), [1648](#), [1651](#), [1652](#), [1657](#)–[1662](#),  
     [1712](#), [1713](#), [1719](#), [1720](#), [1723](#), [1724](#), [1729](#)–  
     [1733](#), [2265](#), [2266](#), [2268](#), [2521](#), [2537](#), [2545](#),  
     [2547](#), [2557](#), [2559](#), [2567](#), [2568](#), [2573](#), [2577](#),  
     [2578](#), [2583](#), [2585](#), [2591](#), [2593](#), [2596](#), [2600](#),  
     [2604](#), [2608](#), [2721](#), [2724](#), [2726](#), [2775](#), [2780](#),  
     [2782](#), [2785](#), [2787](#), [2793](#), [3778](#)–[3780](#), [3785](#)–[3787](#)  
 \MT@char@ ..... [1284](#), [1285](#),  
     [1288](#), [1323](#), [2521](#), [2528](#), [2532](#), [2537](#), [2586](#),  
     [2657](#), [2659](#), [2665](#), [2666](#), [2668](#), [2681](#), [2682](#),  
     [2685](#), [2686](#), [2689](#), [2690](#), [2694](#), [2696](#), [2725](#),  
     [2744](#), [2747](#), [2751](#), [2757](#), [2764](#), [2766](#), [2783](#), [2800](#)  
 \MT@charstring ..... [2534](#), [2724](#), [2729](#), [2756](#)  
 \MT@charxstring ..... [2759](#), [2787](#)  
 \MT@check@active@set .....  
     [4194](#), [4418](#), [4543](#), [4575](#), [4595](#), [4622](#)  
 \MT@check@font ..... [988](#), [2917](#), [2985](#)  
 \MT@check@font@cx ..... [2922](#), [2985](#)  
 \MT@check@range ..... [3911](#), [3913](#)  
 \MT@check@range@ ..... [3913](#), [3914](#)  
 \MT@check@rlist ..... [3860](#), [3904](#)  
 \MT@check@rlist@ ..... [3904](#), [3905](#)  
 \MT@check@step ..... [4531](#)  
 \MT@checklist@ ..... [1075](#), [1094](#), [2250](#)  
 \MT@checklist@family ..... [1112](#)  
 \MT@checklist@font ..... [1156](#)  
 \MT@checklist@size ..... [1140](#)  
 \MT@checksetup ..... [4224](#), [4235](#), [4242](#), [4261](#), [4297](#)  
 \MT@clear@options ..... [161](#), [289](#), [767](#), [4156](#), [4366](#)  
 \MT@clist@break ..... [586](#), [1107](#), [1135](#), [1151](#), [1165](#)  
 \MT@clist@function ..... [586](#)  
 \MT@cnt@encoding ..... [3799](#), [3807](#), [3808](#)  
 \MT@cnt@family ..... [3805](#), [3814](#), [3815](#)  
 \MT@cnt@series ..... [3812](#), [3821](#), [3822](#)  
 \MT@cnt@shape ..... [3819](#), [3827](#), [3828](#)  
 \MT@config@file ..... [4172](#), [4181](#), [4182](#), [4186](#), [4187](#), [4190](#)  
 \MT@context ..... [2442](#), [2454](#), [2473](#), [2500](#)  
 \MT@copy@font ..... [975](#),  
     [1011](#), [3583](#), [3589](#), [3601](#), [3608](#), [4074](#), [4076](#), [4402](#)  
 \MT@copy@font@ ..... [1011](#), [3583](#), [3589](#), [3601](#), [3608](#), [4074](#)  
 \MT@count ..... [503](#), [504](#),  
     [716](#), [1270](#), [1278](#), [1280](#), [1282](#), [1286](#), [1288](#),  
     [1291](#), [1297](#), [1302](#), [1303](#), [1307](#), [1318](#), [1349](#), [1411](#)  
 \MT@curr@file ..... [2362](#), [2363](#), [2372](#), [2373](#),  
     [3338](#), [3339](#), [3557](#), [3559](#), [3562](#), [3750](#), [4186](#), [4765](#)  
 \MT@curr@list@name ..... [1316](#), [1352](#), [1442](#), [1571](#),  
     [2262](#), [2343](#), [2511](#), [2795](#), [2803](#), [2808](#), [2816](#), [2822](#)  
 \MT@curr@ls ..... [1845](#), [1958](#), [1961](#)  
 \MT@curr@set@name ..... [3049](#), [3051](#)–  
     [3053](#), [3055](#), [3058](#), [3059](#), [3064](#), [3069](#), [3073](#),  
     [3074](#), [3083](#), [3108](#), [3112](#), [3141](#), [3171](#), [3177](#),  
     [3183](#), [3187](#), [3188](#), [3439](#), [3515](#), [3536](#), [3551](#),  
     [3570](#), [3642](#), [3646](#), [3653](#), [3656](#), [3658](#), [3664](#),  
     [3667](#), [3670](#), [3682](#), [3696](#), [3704](#), [3717](#), [3751](#), [3753](#)  
 \MT@declare@char@inh ..... [3733](#), [3738](#), [3748](#)  
 \MT@declare@sets ..... [3032](#), [3037](#), [3050](#), [3359](#)  
 \MT@DeclareMicrotypeAlias ..... [3312](#), [3314](#)  
 \MT@DeclareMicrotypeSetDefault ..... [3267](#), [3269](#)  
 \MT@DeclareSet ..... [3028](#), [3030](#), [3046](#)  
 \MT@DeclareSetAndUseIt ..... [3027](#), [3045](#)  
 \MT@DeclareVariants ..... [3299](#), [3300](#), [3302](#)  
 \MT@def@bool@opt ..... [4028](#),  
     [4041](#), [4042](#), [4046](#), [4060](#), [4072](#), [4082](#), [4094](#)  
 \MT@def@n ..... [384](#), [3020](#), [3021](#)  
 \MT@default@ex@set ..... [3284](#)  
 \MT@default@kn@set ..... [3284](#)  
 \MT@default@pr@set ..... [3284](#)  
 \MT@default@sp@set ..... [3284](#)  
 \MT@default@tr@set ..... [3284](#)  
 \MT@define@code@key .....  
     .. [3483](#), [3543](#), [3545](#), [3546](#), [3760](#), [3762](#), [3763](#)  
 \MT@define@code@key@family ..... [3494](#), [3544](#), [3761](#)  
 \MT@define@code@key@font ..... [3520](#), [3548](#), [3765](#)  
 \MT@define@code@key@size ..... [3508](#), [3547](#), [3764](#)  
 \MT@define@opt@key .. [3549](#), [3572](#)–[3575](#), [3713](#)–[3715](#)  
 \MT@define@optionX ..... [4212](#), [4270](#), [4273](#)  
 \MT@define@optionX@ ..... [4287](#), [4316](#), [4318](#), [4319](#)  
 \MT@define@set@key@ ..... [3062](#), [3227](#)–[3230](#)  
 \MT@define@set@key@font ..... [3175](#), [3232](#)  
 \MT@define@set@key@size ..... [3101](#), [3231](#)  
 \MT@detokenize@c ..... [427](#), [2529](#), [2722](#)  
 \MT@detokenize@n ..... [427](#), [2777](#)  
 \MT@dimen@six .....  
     .. [1200](#), [1271](#), [1307](#), [1349](#), [2042](#), [2045](#), [2215](#)  
 \MT@dinfo ..... [87](#)  
 \MT@dinfo@list .....  
     [1092](#), [1102](#), [1105](#), [1110](#), [1119](#), [1122](#), [1128](#),  
     [1130](#), [1138](#), [1146](#), [1149](#), [1154](#), [1164](#), [1167](#), [1171](#)  
 \MT@dinfo@nl ..... [87](#)  
 \MT@DisableLigatures ..... [3349](#)  
 \MT@do@font ..... [677](#), [1217](#), [1535](#), [1670](#), [1740](#), [1837](#)  
 \MT@documentfalse ..... [338](#)  
 \MT@documenttrue ..... [338](#), [4709](#)  
 \MT@dofalse ... [1067](#), [1079](#), [1106](#), [1123](#), [1150](#), [1168](#)  
 \MT@dofont@function ..... [681](#), [698](#)  
 \MT@dotrue [1067](#), [1070](#), [1103](#), [1120](#), [1129](#), [1147](#), [2245](#)  
 \MT@draftfalse ..... [302](#), [4096](#)  
 \MT@drafttrue ..... [302](#), [4098](#)

- \MT@edefn . . . . . [386](#), [2478](#), [2504](#), [3010](#),  
[3489](#), [3503](#), [3559](#), [3562](#), [3565](#), [3749](#), [3891](#), [4128](#)
- \MT@encoding . . . . . [1045](#), [1160](#), [1181](#), [1183](#), [1186](#),  
[2437](#), [2449](#), [2529](#), [2722](#), [2776](#), [2807](#), [2814](#), [2822](#)
- \MT@end@catcodes . . . . . [2360](#),  
[2403](#), [3043](#), [3251](#), [3282](#), [3308](#), [3326](#), [3344](#),  
[3361](#), [3397](#), [3421](#), [3440](#), [3456](#), [3470](#), [3743](#), [4185](#)
- \MT@endinput . . . . . [165](#), [4151](#), [4160](#)
- \MT@engine . . . . . [173](#),  
[270](#), [274](#), [280](#), [286](#), [2146](#), [2150](#), [4152](#), [4158](#)
- \MT@engine@tooold . . . . . [173](#), [271](#)
- \MT@error . . . . . [76](#), [759](#), [872](#),  
[2146](#), [2157](#), [2335](#), [2348](#), [3139](#), [3260](#), [3289](#),  
[3367](#), [3592](#), [3618](#), [3625](#), [4084](#), [4247](#), [4265](#),  
[4305](#), [4475](#), [4487](#), [4505](#), [4562](#), [4633](#), [4646](#), [4726](#)
- \MT@error@doesnt@work . . . . . [4629](#)
- \MT@ex@cc@name . . . . . [1503](#),  
[1505](#), [1577](#), [1578](#), [1588](#), [1594](#), [1595](#), [1607](#), [3405](#)
- \MT@ex@context . . . . . [1016](#), [2977](#), [3019](#)
- \MT@ex@doc@contexts . . . . . [3019](#)
- \MT@ex@factor . . . . . [307](#), [1522](#), [1581](#)
- \MT@ex@factor@ . . . . . [1522](#), [1537](#), [1542](#), [1551](#), [1552](#), [1575](#)
- \MT@ex@inh@name . . . . . [1563](#)–[1565](#)
- \MT@ex@level . . . . . [307](#), [4526](#), [4528](#)
- \MT@ex@max . . . . . [319](#), [1554](#), [1555](#)
- \MT@ex@min . . . . . [319](#), [1557](#), [1558](#)
- \MT@ex@setname . . . . . [3253](#)
- \MT@ex@split@val . . . . . [1549](#)
- \MT@exp@cs . . . . . [381](#), [384](#), [387](#), [390](#), [395](#), [1027](#),  
[1029](#), [1038](#), [1144](#), [1262](#), [1336](#), [1339](#), [1369](#),  
[1373](#), [1383](#), [1386](#), [1389](#), [1565](#), [1652](#), [1724](#),  
[1886](#)–[1888](#), [2465](#), [2940](#), [2948](#), [2949](#), [2953](#),  
[2996](#), [3007](#), [3107](#), [3514](#), [3534](#), [3787](#), [3862](#), [3909](#)
- \MT@exp@gcs . . . . . [381](#), [385](#), [389](#), [391](#), [396](#)
- \MT@exp@one@n . . . . . [398](#), [432](#), [593](#),  
[979](#), [981](#), [1039](#), [1098](#), [1161](#), [1767](#), [2367](#),  
[2917](#), [2919](#), [2925](#), [2937](#), [2959](#), [2993](#), [3004](#),  
[3019](#), [3037](#), [3226](#), [3245](#), [3276](#), [3331](#), [3738](#), [4714](#)
- \MT@exp@two@c . . . . . [400](#), [418](#), [423](#), [433](#), [628](#),  
[978](#), [1020](#), [1022](#), [1024](#), [1033](#), [1805](#), [1935](#),  
[1970](#), [1973](#), [1976](#), [2526](#), [2533](#), [2723](#), [2785](#), [2786](#)
- \MT@exp@two@n . . . . . [402](#), [1116](#), [1125](#), [3068](#), [3182](#)
- \MT@expandfont . . . . . [1507](#), [1525](#), [1529](#)
- \MT@expansion . . . . . [995](#), [1494](#), [4553](#)
- \MT@expansionfalse . . . . . [298](#), [4429](#), [4520](#)
- \MT@expansiontrue . . . . . [298](#), [4166](#)
- \MT@extra@context . . . . . [3019](#), [3389](#), [3405](#), [3428](#),  
[3448](#), [3462](#), [3529](#), [3532](#), [3533](#), [3535](#), [3576](#),  
[3584](#), [3590](#), [3602](#), [3607](#), [3724](#), [3859](#), [3863](#),  
[3866](#), [3869](#), [3870](#), [3875](#), [3880](#), [3881](#), [3883](#), [3910](#)
- \MT@extra@inputenc . . . . . [3722](#), [3747](#), [3752](#), [3753](#)
- \MT@factor@default . . . . . [331](#), [4135](#), [4415](#)
- \MT@family . . . . . [979](#), [1045](#),  
[1117](#), [1160](#), [1181](#), [1183](#), [1186](#), [2438](#), [3322](#), [3323](#)
- \MT@familyalias . . . . . [980](#), [981](#), [1056](#), [1124](#), [1126](#), [2448](#), [2450](#), [3324](#)
- \MT@feat . . . . . [1067](#), [1205](#), [1243](#), [1331](#), [1333](#),  
[1335](#), [1336](#), [1338](#), [1339](#), [1345](#), [1347](#), [1350](#),  
[1358](#)–[1362](#), [1364](#), [1366](#)–[1370](#), [1373](#), [1374](#),  
[1379](#), [1383](#), [1386](#), [1389](#), [1392](#)–[1394](#), [1403](#),  
[1404](#), [1409](#), [1427](#), [1939](#), [2333](#), [2335](#), [2338](#),  
[2339](#), [2343](#), [2345](#), [2348](#), [2474](#), [2476](#), [2478](#),  
[2482](#), [2486](#), [2490](#), [2491](#), [2494](#), [2502](#), [2504](#),  
[2505](#), [2507](#), [2514](#), [2518](#), [2796](#), [2797](#), [2826](#), [2828](#)
- \MT@features . . . . . [752](#), [3019](#), [3032](#), [3226](#), [3240](#), [3271](#), [3554](#), [3733](#)
- \MT@features@long . . . . . [752](#), [755](#), [760](#), [2990](#), [3745](#)
- \MT@file@list . . . . . [2354](#),  
[2356](#), [2364](#), [2367](#), [2369](#), [2374](#), [2377](#), [3331](#), [3335](#)
- \MT@find@file . . . . . [979](#), [981](#), [2354](#)
- \MT@fix@catcode . . . . . [5](#)
- \MT@fix@font@set . . . . . [3091](#), [4386](#)
- \MT@font . . . . . [397](#), [712](#), [853](#), [889](#),  
[978](#), [1033](#), [1040](#), [1201](#), [1208](#), [1215](#), [1216](#),  
[1251](#), [1252](#), [1257](#), [1258](#), [1278](#), [1279](#), [1282](#),  
[1285](#), [1288](#), [1313](#), [1318](#), [1323](#), [1417](#), [1420](#),  
[1455](#), [1456](#), [1458](#), [1459](#), [1531](#), [1535](#), [1561](#),  
[1562](#), [1602](#), [1604](#), [1633](#), [1634](#), [1640](#), [1641](#),  
[1647](#), [1648](#), [1657](#)–[1659](#), [1661](#), [1662](#), [1667](#)–  
[1669](#), [1712](#), [1713](#), [1719](#), [1720](#), [1729](#), [1730](#),  
[1732](#), [1733](#), [1738](#), [1739](#), [1767](#), [1771](#), [1843](#),  
[1844](#), [2254](#), [2544](#), [2599](#), [2830](#), [2886](#)–[2888](#),  
[2890](#), [2905](#), [2906](#), [2917](#), [2918](#), [2925](#), [2951](#), [2960](#)
- \MT@font@copy . . . . . [1016](#), [1024](#), [1025](#), [1033](#)–[1035](#)
- \MT@font@list . . . . . [2830](#), [2917](#)–[2920](#), [2984](#)
- \MT@font@orig . . . . . [1018](#)
- \MT@font@sets . . . . . [3072](#), [3091](#), [3186](#), [4386](#)
- \MT@fontspec@setup . . . . . [836](#), [884](#)
- \MT@fontspec@false . . . . . [832](#), [835](#)
- \MT@fontspec@true . . . . . [832](#), [834](#), [884](#)
- \MT@gdefn . . . . . [384](#), [3380](#), [3396](#),  
[3420](#), [3455](#), [3469](#), [3658](#), [3682](#), [3696](#), [3755](#), [4202](#)
- \MT@get@axis . . . . . [3200](#)–[3203](#), [3213](#)
- \MT@get@basefamily . . . . . [2366](#), [2404](#)
- \MT@get@basefamily@ . . . . . [2410](#), [2413](#)
- \MT@get@char@unit . . . . . [1242](#), [1381](#), [1407](#), [1499](#)
- \MT@get@charwd . . . . . [1276](#), [1307](#), [1384](#), [1407](#)
- \MT@get@config . . . . . [4172](#)
- \MT@get@ex@opt . . . . . [1498](#), [1516](#), [1575](#)
- \MT@get@ex@opt@ . . . . . [1583](#)–[1585](#), [1587](#), [1593](#)
- \MT@get@font . . . . . [3181](#), [3191](#)
- \MT@get@font@ . . . . . [3192](#), [3197](#), [3541](#)
- \MT@get@font@and@size . . . . . [3525](#), [3540](#)
- \MT@get@font@dimen . . . . . [1311](#), [1387](#)
- \MT@get@font@dimen@six . . . . . [1189](#), [1200](#), [1617](#), [1696](#), [1783](#)
- \MT@get@highlevel . . . . . [3067](#), [3077](#), [3215](#), [3488](#), [3499](#)
- \MT@get@inh@list . . . . . [1192](#), [1501](#), [1620](#), [1699](#), [2500](#)
- \MT@get@listname . . . . . [2419](#), [2476](#), [2502](#)
- \MT@get@listname@ . . . . . [2419](#)
- \MT@get@ls@basefont . . . . . [1798](#), [1960](#), [1967](#)
- \MT@get@opt . . . . . [1190](#), [1356](#), [1618](#), [1697](#)
- \MT@get@range . . . . . [3105](#), [3115](#), [3512](#)
- \MT@get@size . . . . . [3122](#), [3127](#), [3136](#), [3155](#), [3211](#)
- \MT@get@slot . . . . . [1239](#), [2264](#), [2521](#), [3777](#), [3784](#)
- \MT@get@slot@ . . . . . [2538](#), [2542](#)
- \MT@get@space@unit . . . . .  
.. [1381](#), [1483](#), [1631](#), [1638](#), [1645](#), [1710](#), [1717](#)
- \MT@get@tr@opt . . . . . [1785](#), [1905](#)
- \MT@get@tr@opt@ . . . . . [1923](#)–[1926](#), [1928](#)
- \MT@get@unit . . . . . [1389](#), [1397](#), [1916](#)
- \MT@get@unit@ . . . . . [1397](#)
- \MT@getkey . . . . . [4353](#), [4368](#)
- \MT@glet . . . . . [379](#), [391](#), [887](#), [1020](#), [1887](#), [1970](#), [2001](#),  
[2151](#), [2159](#), [2884](#), [2968](#), [2985](#)–[2988](#), [3324](#),  
[3583](#), [3601](#), [3802](#), [4074](#), [4076](#), [4677](#), [4683](#), [4769](#)
- \MT@glet@nc . . . . . [390](#), [1186](#), [2014](#), [2206](#), [2514](#),  
[2906](#), [2984](#), [3055](#), [3058](#), [3064](#), [3177](#), [3413](#),  
[3642](#), [3646](#), [3653](#), [3656](#), [3664](#), [3667](#), [3670](#), [3704](#)
- \MT@glet@nn . . . . . [395](#), [3003](#), [4199](#)

- \MT@if@false ..... [296](#),  
     854, 861, 891, 908, 2930, 3908, 4745, 4752  
 \MT@if@fontspec@font .....  
     .... 680, [836](#), 1800, 1835, 1844, 2267, 2280  
 \MT@if@list@exists .....  
     .. 1179, 1497, 1515, 1616, 1695, 1784, [2473](#)  
 \MT@if@outer@next .....  
     .. [2085](#), 2100, 2104, 2107, 2112, 2117–2120  
 \MT@if@true ..... [296](#),  
     855–857, 862–864, 892–894, 921, 925, 926,  
     2923, 3917, 3921, 3929, 3934, 4746–4750, 4753  
 \MT@if@defined@cc@T ..... [406](#), 858,  
     897, 1260, 1296, 1563, 1650, 1722, 1829,  
     2709, 3093, 3322, 3752, 3849, 4208, 4211, 4772  
 \MT@if@defined@cc@TF .....  
     ..... [406](#), 794, 1788, 1813, 1816, 1827,  
     1958, 2039, 2212, 2259, 2260, 2477, 2503, 3858  
 \MT@if@defined@n@T .....  
     . [406](#), 783, 1095, 1113, 1141, 1157, 1261,  
     1392, 1428, 1564, 1588, 1651, 1723, 1908,  
     1910, 1929, 2456, 2825, 3052, 3317, 3408,  
     3566, 3808, 3815, 3822, 3828, 3859, 3869, 4353  
 \MT@if@defined@n@TF .....  
     [406](#), 1056, 1073, 1096, 1114, 1142, 1158,  
     1181, 1358, 1366, 1577, 1594, 1885, 2248,  
     2338, 2444, 2529, 3080, 3254, 3257, 3285,  
     3557, 3835, 3890, 3899, 3969, 4195, 4198, 4712  
 \MT@if@dim .. [526](#), 648, 649, 653, 654, 3138, 3147,  
     3915–3917, 3919, 3920, 3927–3929, 3932, 3933  
 \MT@if@dimen .... [488](#), 3166, 3645, 3655, 3669, 4144  
 \MT@if@empty ..... [439](#), 1215, 1216, 1249, 1255,  
     1472, 1473, 1630, 1637, 1644, 1667–1669,  
     1681–1683, 1685–1687, 1709, 1716, 1738,  
     1739, 1755, 1756, 2020, 2021, 2031, 2049,  
     2181, 2204, 2205, 2217, 2995, 3031, 3035,  
     3079, 3089, 3116, 3117, 3130, 3131, 3205,  
     3216, 3239, 3243, 3270, 3274, 3356, 3435,  
     3550, 3556, 3576, 3582, 3590, 3600, 3607,  
     3680, 3732, 3737, 3982, 4013, 4175, 4221, 4294  
 \MT@if@fint ..... [452](#), 2688, 3436, 3681, 4127, 4137  
 \MT@if@streq ..... [535](#), 1085, 1434,  
     2334, 2475, 2490, 2745, 2748, 2756, 2765,  
     2958, 3180, 3524, 3644, 3666, 3668, 3850,  
     3870, 3941, 3985, 3987, 3990, 3993, 4015,  
     4017, 4031, 4032, 4104, 4105, 4110, 4114,  
     4143, 4223, 4230, 4234, 4241, 4296, 4302, 4605  
 \MT@in@clist ..... [614](#),  
     755, 784, 1098, 1116, 1125, 1161, 1767,  
     2356, 2367, 2917, 2919, 2925, 2937, 2993, 3331  
 \MT@in@rlist ..... [642](#), 1144, 2465  
 \MT@in@rlist@ ..... [642](#)  
 \MT@in@rlist@ ..... [642](#)  
 \MT@in@tlist ..... [630](#), 2416, 3004  
 \MT@in@tlist@ ..... [630](#)  
 \MT@in@not@false ..... [100](#)  
 \MT@in@not@true ..... [100](#)  
 \MT@in@increment ..... [716](#), 3807, 3814, 3821, 3827  
 \MT@info ..... [76](#), 91, 96, 4111, 4337, 4341  
 \MT@info@missing@char ..... 1291, [1321](#), 2595  
 \MT@info@nl ..... [76](#), 92, 93, 97, 795, 1182,  
     1322, 4102, 4112, 4182, 4196, 4200, 4398,  
     4402, 4414, 4421, 4527, 4554, 4574, 4579,  
     4588, 4598, 4621, 4625, 4638, 4653, 4722, 4733  
 \MT@info@not@tracking ..... 993, 1086, [1090](#)  
 \MT@info@not@tracking@ ..... [1086](#), [1090](#)  
 \MT@inh@do ..... 2513, [3766](#)  
 \MT@inh@feat ..... [3722](#), 3732, 3735, 3746  
 \MT@inh@split ..... 3768, [3772](#)  
 \MT@in@list@false ..... [614](#), 618, 631, 643, 2933  
 \MT@in@list@true ..... [614](#), 620, 638, 650, 655, 2933  
 \MT@is@active ..... 2526, [2701](#)  
 \MT@is@char ..... 2533, 2723, [2729](#)  
 \MT@is@charx ..... 2738, 2759, 2786  
 \MT@is@composite ..... 2531, [2773](#)  
 \MT@is@feature ..... [754](#), 3036, 3244, 3275  
 \MT@is@letter ..... 2527, [2645](#), 2726, 2782  
 \MT@is@number ..... 2671, [2676](#)  
 \MT@is@symbol ..... 2530, [2720](#)  
 \MT@is@uni@comp ..... 2780, 2792, [2792](#)  
 \MT@is@xchar ..... 2749, [2755](#)  
 \MT@iterate ..... 664  
 \MT@kerning ..... 999, [1692](#), 4624  
 \MT@kerning@false ..... [304](#)  
 \MT@kerning@true ..... [304](#)  
 \MT@kn@cc@name ..... 1701, 1703, 1746, 3462  
 \MT@kn@context ..... 2978, [3019](#)  
 \MT@kn@doc@contexts ..... [3019](#)  
 \MT@kn@factor ..... [307](#)  
 \MT@kn@factor@ ..... [1358](#)  
 \MT@kn@inh@name ..... 1722–1724  
 \MT@kn@max ..... [319](#)  
 \MT@kn@min ..... [319](#)  
 \MT@kn@set@name ..... [3253](#)  
 \MT@kn@split@val ..... [1707](#)  
 \MT@kn@unit ..... [313](#)  
 \MT@kn@unit@ ..... [1366](#), 1749  
 \MT@led@unhbox@line ..... [795](#)  
 \MT@led@mac@setup ..... [789](#), 878–880  
 \MT@let@cn [392](#), 395, 396, 839, 1057, 1196, 1505,  
     1578, 1624, 1703, 1909, 1911, 1915, 2333,  
     2345, 2470, 2474, 2507, 2886, 3159, 3570, 3751  
 \MT@let@enc .. 390, 1394, 2126, 2486, 2518, 2928,  
     3892, 3897, 3991, 3994, 4237, 4244, 4636, 4651  
 \MT@let@nn .....  
     [395](#), 1359, 1364, 1367, 1379, 1595, 1598, 1930  
 \MT@letterspace .....  
     .... [334](#), 1788, 1909, 1919, 4686, 4687, 4689  
 \MT@letterspace@ ..... 1226, 1296,  
     1297, 1788, 1789, 1792, 1795, 1803, 1810,  
     1815, 1823, 1836–1838, 1845, 1854, 1898–  
     1901, 2045, 2049, [2167](#), 2182–2185, 2218, 2221  
 \MT@letterspace@default ..... [334](#), 4687  
 \MT@list@name ..... 2419,  
     2467, 2470, 2477, 2478, 2480, 2482, 2503,  
     2504, 2506, 2507, 2511, 2514, 3787, 3792  
 \MT@list@name@count ..... 3553, 3558, 3560  
 \MT@load@input@enc ..... 1435, 1438, 1447  
 \MT@load@list ..... 1194, 1503, 1622, 1701, 2330  
 \MT@loop ..... 664, 672, 710, 3896  
 \MT@lower ..... 3109, [3115](#), 3515  
 \MT@ls@adjust ..... 1852, [2178](#)  
 \MT@ls@adjust@ ..... [2162](#), 2186  
 \MT@ls@adjust@empty ..... 2164, [2178](#)  
 \MT@ls@adjust@relax ..... 2165, [2178](#)  
 \MT@ls@basefont ..... 1967, 1976, 1979, 1980  
 \MT@ls@fontspec@colon ..... 1894  
 \MT@ls@fontspec@font ..... 1805, [1894](#)  
 \MT@ls@outer@k ..... 1855,  
     1860, 1868, 2062, 2076, 2123, 2141, [2226](#)  
 \MT@ls@set@ls ..... 2169, 2172, [2178](#)

- `\MT@ls@too@large` ..... 2185, 2188, 4689  
`\MT@lsfont` ..... 1794, 1804, 1810,  
1823, 1824, 1833, 1837, 1838, 1843, 1935,  
1945, 1990, 1992, 2007, 2008, 2015, 2016,  
2022, 2026, 2033, 2043, 2047, 2050, 2219, 2222  
`\MT@ltx@pickupfont` ..... 914, 958, 963, 2901, 2909  
`\MT@lua` 251, 357, 476, 513, 558, 682, 2268, 2281, 2549  
`\MT@luatex@no` ..... 249, 347  
`\MT@map@clist@` ..... 586  
`\MT@map@clist@c` ..... 586,  
1026, 2263, 2924, 2936, 2947, 2983, 2990,  
3032, 3240, 3271, 3554, 3733, 3735, 3745, 4351  
`\MT@map@clist@n` .....  
586, 1072, 2247, 3019, 3034, 3054, 3065,  
3103, 3178, 3226, 3242, 3273, 3303, 3378,  
3486, 3497, 3510, 3522, 3662, 3678, 3781,  
3977, 3980, 4009, 4011, 4040, 4122, 4219, 4292  
`\MT@map@tlist@` ..... 604  
`\MT@map@tlist@c` .... 604, 633, 644, 1029, 1262,  
1565, 1652, 1724, 2423, 2953, 3860, 3909, 4386  
`\MT@map@tlist@n` ..... 604, 2395, 2731, 3801  
`\MT@max@char` ..... 2645, 2668  
`\MT@max@slot` ..... 2645, 2694  
`\MT@maybe@do` .. 1067, 1176, 1495, 1614, 1693, 1769  
`\MT@maybe@gobble@with@tikz` ..... 843, 1884, 1938  
`\MT@maybe@rem@from@list` ..... 2953, 2957  
`\MT@minus` ..... 1803, 1898–1901  
`\MT@MT` ..... 2, 76, 79, 82, 84,  
86, 91, 92, 94, 95, 285, 766, 873, 875, 3592,  
4154, 4176, 4337, 4341, 4399, 4609, 4726, 4727  
`\MT@MT@pickupfont` ..... 960, 965, 2901, 2911  
`\MT@next@listname` ..... 2426, 2435  
`\MT@next@listname@` ..... 2445, 2457, 2463  
`\MT@n@ligatures` ..... 2254, 3349  
`\MT@n@setname` ..... 2246, 3349  
`\MT@nofamily@false` ..... 1173, 1178  
`\MT@nofamily@true` ..... 1173, 2436  
`\MT@n@ligatures` ..... 1003, 2242, 4698  
`\MT@n@ligatures@` ..... 1990, 1992, 2254, 2257  
`\MT@n@ligatures@false` ..... 301  
`\MT@n@ligatures@true` ..... 301, 3358  
`\MT@nonselected@false` ..... 1511, 1526  
`\MT@nonselected@true` ..... 1511, 1514  
`\MT@norest@false` .....  
.. 2654, 2661, 2670, 2691, 2745, 2757, 2765  
`\MT@norest@true` ..... 2524, 2654, 2679  
`\MT@old@cmd` ..... 66, 71–74  
`\MT@opt@autof@false` ..... 3952  
`\MT@opt@autof@true` ..... 3952, 4042  
`\MT@opt@def@set` ..... 3968, 3996, 4021  
`\MT@opt@DVIf@false` ..... 3953  
`\MT@opt@DVIf@true` ..... 3953, 4049, 4052  
`\MT@opt@expansion@false` ..... 3951  
`\MT@opt@expansion@true` ..... 3951  
`\MT@options` ..... 3473  
`\MT@optwarn@admissible` ..... 3954, 4033, 4114  
`\MT@optwarn@nan` ..... 3958, 4129, 4139  
`\MT@orig@add@accent` ..... 2907  
`\MT@orig@foreign@language` ..... 4739, 4741  
`\MT@orig@pickupfont` ..... 2836, 2903  
`\MT@orig@py@macron` ..... 955, 959, 964  
`\MT@orig@select@language` ..... 4734, 4736  
`\MT@outer@kern` 1854, 1858, 1860, 1861, 1865, 1873,  
1961, 1962, 1964, 2123, 2124, 2141, 2199, 2238  
`\MT@outer@space` ..... 1848,  
1849, 2005, 2058, 2060, 2061, 2074, 2075,  
2092, 2093, 2108, 2109, 2113, 2114, 2139  
`\MT@pdf@annot` ..... 99  
`\MT@pdf@or@lua` ..... 178, 186, 283, 455, 4670  
`\MT@pdf@tex@no` ..... 208, 346  
`\MT@permute` 3395, 3419, 3433, 3454, 3468, 3758, 3797  
`\MT@permute@` ..... 3797  
`\MT@permute@@` ..... 3797  
`\MT@permute@@@` ..... 3797  
`\MT@permute@@@@` ..... 3797  
`\MT@permute@@@@@` ..... 3826, 3831  
`\MT@permute@@@@@@` ..... 3839, 3841  
`\MT@permute@define` ..... 3832, 3842–3844, 3888  
`\MT@permute@reset` ..... 3801, 3894  
`\MT@permutelist` ..... 3389, 3405,  
3431, 3448, 3462, 3529, 3530, 3535, 3756,  
3854, 3855, 3859, 3863, 3866, 3869–3871,  
3873–3875, 3883, 3884, 3910, 3942, 3944, 3945  
`\MT@pickupfont` ..... 2901  
`\MT@plain` ..... 125, 146  
`\MT@pr@c@name` ..... 1194, 1196, 1463, 3389  
`\MT@pr@context` ..... 1016, 2977, 3019  
`\MT@pr@doc@contexts` ..... 3019  
`\MT@pr@factor` ..... 307, 4138, 4415, 4416  
`\MT@pr@factor@` ..... 1358  
`\MT@pr@inh@name` ..... 1260, 1261, 1263  
`\MT@pr@level` ..... 307, 4413, 4414  
`\MT@pr@max` ..... 319  
`\MT@pr@min` ..... 319  
`\MT@pr@setname` ..... 3253  
`\MT@pr@split@val` ..... 1247  
`\MT@pr@unit` ..... 313, 4145, 4417  
`\MT@pr@unit@` ..... 1366, 1466  
`\MT@preset@aux` ..... 1468,  
1470, 1472, 1473, 1476, 1751, 1753, 1755, 1756  
`\MT@preset@aux@factor` 1468, 1476, 1681–1683, 1751  
`\MT@preset@aux@space` . 1470, 1476, 1685–1687, 1753  
`\MT@preset@ex` ..... 1589, 1606  
`\MT@preset@kn` ..... 1744  
`\MT@preset@kn@` ..... 1744  
`\MT@preset@pr` ..... 1461  
`\MT@preset@pr@` ..... 1461  
`\MT@preset@sp` ..... 1674  
`\MT@preset@sp@` ..... 1674  
`\MT@ProcessOptionsWithKV` ..... 4345, 4369  
`\MT@protrude@chars` ..... 4270, 4274, 4413, 4576  
`\MT@protrusion` ..... 994, 1175, 4420  
`\MT@protrusion@false` ..... 297  
`\MT@protrusion@true` ..... 297, 4161  
`\MT@rbba@expansion` ..... 747  
`\MT@rbba@kerning` ..... 747  
`\MT@rbba@protrusion` ..... 747  
`\MT@rbba@spacing` ..... 747  
`\MT@rbba@tracking` ..... 747  
`\MT@register@font` ..... 1007, 2918, 2986  
`\MT@register@font@cx` ..... 2946, 2986  
`\MT@register@subst@font` ..... 2892, 2919, 2987  
`\MT@register@subst@font@cx` ..... 2935, 2987  
`\MT@rem@from@clist` ..... 625, 1039, 2959, 3478  
`\MT@rem@from@list` ..... 1029, 1037  
`\MT@rem@last@space` ..... 427  
`\MT@repeat` ..... 664, 674, 712, 3902  
`\MT@requires@latex` 145, 162, 775, 833, 883, 910,  
941, 1934, 2834, 3960, 4121, 4346, 4350, 4768

- `\MT@requires@luatex` . . . . . [341](#), [985](#), [1013](#), [1224](#),  
[1529](#), [1539](#), [1763](#), [2257](#), [4045](#), [4275](#), [4289](#), [4570](#)  
`\MT@requires@pdftex` . . . . . [341](#),  
[454](#), [489](#), [536](#), [790](#), [974](#), [984](#), [998](#), [1002](#),  
[1014](#), [1223](#), [1268](#), [1294](#), [1538](#), [1613](#), [1692](#),  
[1762](#), [1987](#), [2243](#), [3350](#), [3580](#), [3695](#), [3703](#),  
[4071](#), [4165](#), [4288](#), [4440](#), [4472](#), [4499](#), [4569](#), [4696](#)  
`\MT@res@a` . . . . . [548](#), [550](#), [616](#), [623](#), [626](#), [628](#), [632](#), [637](#)  
`\MT@res@b` . . . . . [549](#), [550](#), [626–628](#), [636](#), [637](#)  
`\MT@reset@context` . . . . . [2967](#), [2971](#), [2975](#), [3002](#)  
`\MT@reset@context@` . . . . . [2975](#), [3002](#)  
`\MT@reset@ef@codes` . . . . . [1500](#), [1524](#), [1538](#), [1590](#)  
`\MT@reset@ef@codes@` . . . . . [1533](#), [1543](#), [1547](#)  
`\MT@reset@kn@codes` . . . . . [1698](#), [1705](#), [1735](#)  
`\MT@reset@kn@codes@` . . . . . [1735](#)  
`\MT@reset@pr@codes` . . . . . [1191](#), [1198](#), [1219](#)  
`\MT@reset@pr@codes@` . . . . . [1219](#)  
`\MT@reset@sp@codes` . . . . . [1619](#), [1626](#), [1664](#)  
`\MT@reset@sp@codes@` . . . . . [1664](#)  
`\MT@restore@catcodes` . . . . . [5](#), [7](#), [8](#), [163](#), [4774](#)  
`\MT@restore@p@h` . . . . . [826](#), [865](#), [927](#)  
`\MT@saved@setupfont` . . . . . [4334](#), [4342](#)  
`\MT@scale` . . . . . [723](#), [1271](#), [1332](#), [1346](#), [1349](#), [1552](#)  
`\MT@scale@factor` . . . . . [1273](#), [1304](#), [1330](#), [1478](#), [1608](#)  
`\MT@scale@to@em` . . . . . [1250](#), [1256](#), [1268](#), [1301](#), [1484](#),  
[1632](#), [1639](#), [1646](#), [1711](#), [1718](#), [1918](#), [2040](#), [2213](#)  
`\MT@scrubfeature` . . . . . [1049](#), [1060](#), [3501](#)  
`\MT@scrubfeatures` . . . . . [1060](#), [3096](#), [3527](#)  
`\MT@selectedfalse` . . . . . [300](#)  
`\MT@selectedtrue` . . . . . [300](#)  
`\MT@series` . . . . . [1045](#), [1160](#), [2439](#), [2451](#)  
`\MT@set@all@ex` . . . . . [1533](#), [1609](#)  
`\MT@set@all@kn` . . . . . [1735](#), [1757](#)  
`\MT@set@all@pr` . . . . . [1212](#), [1219](#), [1474](#)  
`\MT@set@all@sp` . . . . . [1664](#), [1689](#)  
`\MT@set@babel@context` . . . . . [4711](#), [4737](#), [4742](#), [4756](#)  
`\MT@set@codes` . . . . . [1197](#), [1230](#), [1506](#), [1625](#), [1704](#), [2346](#)  
`\MT@set@curr@ls` . . . . . [1845](#)  
`\MT@set@curr@ok` . . . . . [1865](#), [1867](#), [1873](#), [1876](#), [2097](#), [2101](#)  
`\MT@set@curr@os` . . . . . [1847](#), [1875](#), [2097](#), [2101](#)  
`\MT@set@default@set` . . . . . [3271](#), [3276](#), [3284](#)  
`\MT@set@ex@codes` . . . . . [1528](#), [4511](#), [4513](#)  
`\MT@set@ex@codes@n` . . . . . [1511](#), [1528](#), [4513](#)  
`\MT@set@ex@codes@s` . . . . . [1496](#), [4511](#)  
`\MT@set@ex@heirs` . . . . . [1565](#), [1601](#)  
`\MT@set@inh@list` . . . . . [3729](#), [3731](#)  
`\MT@set@inputenc` . . . . . [1193](#), [1425](#), [1502](#), [1621](#), [1700](#), [2512](#)  
`\MT@set@inputenc@` . . . . . [1428](#), [1430](#)  
`\MT@set@kn@codes` . . . . . [1694](#)  
`\MT@set@kn@heirs` . . . . . [1724](#), [1728](#)  
`\MT@set@listname` . . . . . [1195](#),  
[1357](#), [1504](#), [1576](#), [1623](#), [1702](#), [1907](#), [2795](#)  
`\MT@set@ls@basefont` . . . . . [1976](#)  
`\MT@set@ls@font` . . . . . [1842](#), [1933](#), [1945](#)  
`\MT@set@named@keys` . . . . . [3391](#), [3407](#), [3429](#), [3450](#), [3464](#), [3473](#)  
`\MT@set@pr@codes` . . . . . [1177](#)  
`\MT@set@pr@heirs` . . . . . [1264](#), [1453](#)  
`\MT@set@sp@codes` . . . . . [1615](#)  
`\MT@set@sp@heirs` . . . . . [1652](#), [1656](#)  
`\MT@set@tr@codes` . . . . . [1779](#), [1940](#), [1950](#)  
`\MT@set@tr@zero` . . . . . [1790](#), [1976](#), [4316](#)  
`\MT@SetExpansion` . . . . . [3403](#), [3405](#)  
`\MT@SetExtraKerning` . . . . . [3460](#), [3462](#)  
`\MT@SetExtraSpacing` . . . . . [3446](#), [3448](#)  
`\MT@SetProtrusion` . . . . . [3387](#), [3389](#)  
`\MT@SetTracking` . . . . . [3425](#), [3427](#)  
`\MT@setup@` . . . . . [773](#), [777](#), [4063](#), [4064](#), [4769](#)  
`\MT@setup@contexts` . . . . . [2966](#), [2982](#)  
`\MT@setup@copies` . . . . . [4385](#), [4401](#)  
`\MT@setup@expansion` . . . . . [4388](#), [4425](#)  
`\MT@setup@kerning` . . . . . [4392](#), [4616](#), [4659](#)  
`\MT@setup@no@ligatures` . . . . . [4393](#), [4694](#)  
`\MT@setup@PDF` . . . . . [4384](#), [4396](#), [4406](#)  
`\MT@setup@protrusion` . . . . . [4387](#), [4409](#)  
`\MT@setup@spacing` . . . . . [4391](#), [4583](#), [4660](#)  
`\MT@setup@spacing@check` . . . . . [4601](#), [4772](#), [4773](#)  
`\MT@setup@tracking` . . . . . [4389](#), [4568](#), [4631](#), [4658](#)  
`\MT@setup@warntracking` . . . . . [4390](#), [4663](#)  
`\MT@setupfont` . . . . . [972](#), [2895](#), [4335](#), [4338](#), [4342](#), [4378](#)  
`\MT@setupfont@hook` . . . . .  
. . . . . [852](#), [887](#), [889](#), [896](#), [901](#), [927](#), [934](#), [973](#)  
`\MT@shape` . . . . . [1045](#), [1160](#), [2440](#), [2452](#)  
`\MT@shorthandoff` . . . . . [4720](#), [4751](#), [4754](#)  
`\MT@show@pdfannot` . . . . . [106](#)  
`\MT@shrink` . . . . . [316](#), [1519](#),  
[4436](#), [4437](#), [4444](#), [4445](#), [4448](#), [4452](#), [4516](#), [4529](#)  
`\MT@shrink@` . . . . . [1519](#), [1531](#), [1575](#)  
`\MT@shrink@default` . . . . . [332](#)  
`\MT@size` . . . . . [649](#), [653](#), [654](#), [1045](#), [1160](#)  
`\MT@size@name` . . . . . [642](#), [2467](#)  
`\MT@sp@cname` . . . . . [1622](#), [1624](#), [1676](#), [3448](#)  
`\MT@sp@context` . . . . . [2978](#), [3019](#), [4605](#)  
`\MT@sp@doc@contexts` . . . . . [3019](#)  
`\MT@sp@factor` . . . . . [307](#)  
`\MT@sp@factor@` . . . . . [1358](#)  
`\MT@sp@inh@name` . . . . . [1650–1652](#)  
`\MT@sp@max` . . . . . [319](#)  
`\MT@sp@min` . . . . . [319](#)  
`\MT@sp@setname` . . . . . [3253](#)  
`\MT@sp@split@val` . . . . . [1628](#)  
`\MT@sp@unit` . . . . . [313](#)  
`\MT@sp@unit@` . . . . . [1366](#), [1679](#)  
`\MT@spacing` . . . . . [999](#), [1612](#), [4597](#)  
`\MT@spacingfalse` . . . . . [303](#)  
`\MT@spacingtrue` . . . . . [303](#)  
`\MT@split@codes` . . . . . [1232](#), [1236](#)  
`\MT@split@name` . . . . . [978](#), [1045](#)  
`\MT@step` . . . . . [316](#), [1520](#), [4439](#),  
[4441](#), [4459](#), [4463](#), [4466](#), [4530](#), [4533](#), [4534](#), [4538](#)  
`\MT@step@` . . . . . [1520](#), [1531](#), [1575](#)  
`\MT@stretch` . . . . . [316](#), [1518](#), [4433](#),  
[4434](#), [4437](#), [4444](#), [4446](#), [4451](#), [4454](#), [4515](#), [4529](#)  
`\MT@stretch@` . . . . . [1518](#), [1531](#), [1575](#)  
`\MT@stretch@default` . . . . . [332](#), [4434](#)  
`\MT@strip@prefix` . . . . . [2738](#), [2759](#), [2785](#)  
`\MT@temp` . . . . . [1214–1217](#), [1666–1670](#),  
[1737–1740](#), [2010](#), [2014](#), [2017](#), [2022](#), [2025](#),  
[2030](#), [2089](#), [2092](#), [2098](#), [2102](#), [2105](#), [2108](#),  
[2113](#), [2121](#), [2123](#), [2129](#), [2203](#), [2206](#), [2223](#),  
[3199](#), [3224](#), [4349](#), [4354](#), [4359](#), [4360](#), [4362](#), [4365](#)  
`\MT@tempencoding` . . . . . [3834](#), [3835](#), [3845](#), [3851](#)  
`\MT@tempfamily` . . . . . [3846](#)  
`\MT@tempseries` . . . . . [3847](#)  
`\MT@tempshape` . . . . . [3848](#)  
`\MT@tempsize` . . . . . [3802](#), [3849](#), [3858](#), [3860](#), [3864](#)  
`\MT@test@ast` . . . . . [2038](#), [2211](#), [3078](#), [3087](#)  
`\MT@text@ls` . . . . . [49](#), [2164](#), [2165](#), [2167](#)  
`\MT@the@pr@code` . . . . . [1221](#), [1251](#), [1257](#), [1839](#)  
`\MT@the@pr@code@tr` . . . . . [1221](#), [1839](#)  
`\MT@tikz@setup` . . . . . [843](#), [866](#), [953](#)

- `\MT@tlist@break` . . . . . 604, 639, 661, 2428, 3948  
`\MT@toks` 295, 1322, 1352, 1571, 2525, 2560, 2587,  
2659, 2666, 2713, 2751, 2766, 2800, 2812, 2820  
`\MT@tr@cc@name` . . . . .  
. . . 1908–1911, 1915, 1929, 1930, 3430, 3437  
`\MT@tr@context` . . . . . 2978, 3019  
`\MT@tr@doc@contexts` . . . . . 3019  
`\MT@tr@factor@` . . . . . 1917  
`\MT@tr@font@list` . . . . . 1765  
`\MT@tr@ispace` . . . . . 1813, 1814, 1923  
`\MT@tr@ligatures` . . . . . 1829, 1926, 1989, 1992  
`\MT@tr@max` . . . . . 319, 2189–2191  
`\MT@tr@min` . . . . . 319, 2193–2195  
`\MT@tr@noligatures` . . . . . 1829, 1986  
`\MT@tr@okern` . . . . . 1827, 1925  
`\MT@tr@ospace` . . . . . 1816, 1817, 1923  
`\MT@tr@outer@icr` . . . . . 2102, 2131  
`\MT@tr@outer@icr@` . . . . . 2131  
`\MT@tr@outer@l` . . . . . 1850, 2054  
`\MT@tr@outer@next` . . . . . 2082, 2086, 2133  
`\MT@tr@outer@r` . . . . . 1877, 2082, 2098, 2105, 2131  
`\MT@tr@outer@r@` . . . . . 2083, 2088  
`\MT@tr@set@okern` . . . . . 1828, 2199  
`\MT@tr@set@okern@` . . . . . 2204, 2205, 2210  
`\MT@tr@set@space` . . . . . 1819, 2006  
`\MT@tr@set@space@` . . . . . 2011–2013, 2019  
`\MT@tr@set@space@@` . . . . . 2024, 2029, 2032, 2037  
`\MT@tr@set@name` . . . . . 3253  
`\MT@tr@unit@` . . . . . 1910, 2039, 2212  
`\MT@tracking` 986, 1765, 1940, 2897, 4316, 4317, 4578  
`\MT@tracking@` . . . . . 1765, 4317  
`\MT@tracking@false` . . . . . 305, 4635  
`\MT@tracking@true` . . . . . 305  
`\MT@try@order` . . . . . 2423, 2431  
`\MT@undefined@char` . . . . . 2706, 2708, 2719  
`\MT@upper` . . . . . 3109, 3115, 3515  
`\MT@use@set` . . . . . 3240, 3245, 3253  
`\MT@UseMicrotypeSet` . . . . . 3236, 3238  
`\MT@val` . . . . . 2995–2997, 2999, 3005, 3007,  
3010, 3066, 3070, 3078, 3081, 3084, 3104–  
3106, 3118, 3121, 3123, 3126, 3128, 3129,  
3135, 3137, 3138, 3140, 3143, 3145, 3156,  
3159, 3163, 3166–3168, 3170, 3172, 3179–  
3181, 3184, 3193–3195, 3207, 3209, 3214,  
3216, 3218, 3220, 3487, 3489, 3498, 3501,  
3503, 3511–3513, 3523–3525, 3531, 3536,  
3780, 3787, 3791, 3792, 3969, 3970, 3973,  
3981, 3982, 3985, 3987, 3990, 3993, 4012,  
4013, 4015, 4017, 4220, 4221, 4223, 4230,  
4234, 4241, 4247, 4293, 4294, 4296, 4302, 4305  
`\MT@variants` . . . . . 2416, 3295, 3306  
`\MT@vinfo` . . . . . 76, 93, 992, 1087, 1091, 1361,  
1370, 1374, 1409, 1579, 1596, 1782, 1792,  
2271, 2282, 2339, 2363, 2373, 2376, 2480,  
2482, 2488, 2976, 2999, 3333, 3339, 4102,  
4109, 4226, 4232, 4238, 4245, 4299, 4304, 4713  
`\MT@warn@ascii` . . . . . 2668, 2799  
`\MT@warn@axis@empty` . . . . . 3206, 3217, 3222  
`\MT@warn@code@too@large` . . . . . 1336, 1339, 1343  
`\MT@warn@err` . . . . . 76, 4106, 4107  
`\MT@warn@ex@too@large` . . . . . 1555, 1558, 1569  
`\MT@warn@maybe@inputenc` . . . . . 2261, 2813, 2821, 2824  
`\MT@warn@nodim` . . . . . 3634, 3647, 3657, 3671  
`\MT@warn@number@too@large` . . . . . 2695, 2805  
`\MT@warn@preset@twidth` . . . . . 1467, 1487, 1680, 1750  
`\MT@warn@rest` . . . . . 2566, 2603, 2810  
`\MT@warn@tracking@DVI` . . . . . 1793, 4665  
`\MT@warn@unknown` . . . . . 2556, 2572, 2582, 2607, 2818  
`\MT@warn@unknown@once` . . . . . 2836, 2837  
`\MT@warning` . 67, 76, 94, 2190, 2194, 2274, 2491,  
3053, 3082, 3170, 3223, 3318, 3341, 3438,  
3567, 3636, 3684, 3698, 3708, 3852, 3873,  
3944, 4106, 4111, 4208, 4285, 4324, 4326, 4327  
`\MT@warning@nl` . . . . . 76, 88, 95, 272, 765,  
769, 805, 817, 903, 1202, 1314, 1350, 1402,  
1442, 1488, 1570, 1998, 2800, 2806, 2811,  
2819, 2868, 3410, 3609, 3955, 3962, 3973,  
4055, 4107, 4112, 4146, 4153, 4189, 4203,  
4375, 4464, 4517, 4536, 4590, 4606, 4671, 4758  
`\MT@while@num` . . . . . 670, 677, 683, 930–932  
`\MT@with@babel@and@T` . . . . .  
. . . 782, 855–857, 892–894, 4746–4750, 4753  
`\MT@with@package@T` . . . 779, 830, 834, 859, 862–  
864, 866, 871, 878–881, 884, 899, 911, 925,  
926, 928, 942, 953, 954, 2836, 2837, 2879, 4589  
`\MT@xadd` . . . . . 104, 570, 2364, 2369, 2374,  
2377, 2940, 2949, 3007, 3107, 3335, 3514, 3787  
`\MT@xaddb` . . . . . 578, 3534, 3862  
`\MT@xdef@n` . . 386, 3255, 3258, 3287, 3292, 3321,  
3437, 3529, 3551, 3717, 3753, 3883, 3970, 3972  
`\MT@xetex@no` . . . . . 240  
`\MT@xspace` . . . . . 2121, 2136  
`\MT@xspace@` . . . . . 2136  
`\MT@xunicode@false` . . . . . 829  
`\MT@xunicode@true` . . . . . 829, 830, 881  
  
N `newpx` (package) . . . . . 21, 143, 235  
`newtx` (package) . . . . . 21, 143, 235  
`newunicodechar` (package) . . . . . 94, 235  
  
O `\outputmode` . . . . . 191, 4045  
  
P `\pdfadjustinterwordglue` . . . . . 4318, 4587  
`\pdfadjustspacing` . . . . . 4277, 4281, 4549  
`\pdfappendkern` . . . . . 4320, 4620  
`\pdfcopyfont` . . . . . 1013, 1024  
`pdfcpot` (package) . . . . . 26, 30, 54, 64, 97, 227  
`\pdffontexpand` . . . . . 1529, 1531  
`\pdfnoligatures` . . . . . 2257, 2273, 2279  
`\pdfprependkern` . . . . . 4319, 4619  
`\pdfprotrudechars` . . . . . 192, 4276, 4280, 4682  
`pdfTeX` (engine) 1, 4, 6–9, 13–19, 23–29, 31, 32, 34,  
35, 37–40, 42, 43, 45, 46, 52, 57, 58, 64, 69,  
4013, 4015, 4017, 4220, 4221, 4223, 4230,  
4234, 4241, 4247, 4293, 4294, 4296, 4302, 4305  
`\MT@variants` . . . . . 2416, 3295, 3306  
`\MT@vinfo` . . . . . 76, 93, 992, 1087, 1091, 1361,  
1370, 1374, 1409, 1579, 1596, 1782, 1792,  
2271, 2282, 2339, 2363, 2373, 2376, 2480,  
2482, 2488, 2976, 2999, 3333, 3339, 4102,  
4109, 4226, 4232, 4238, 4245, 4299, 4304, 4713  
`\MT@warn@ascii` . . . . . 2668, 2799  
`\MT@warn@axis@empty` . . . . . 3206, 3217, 3222  
`\MT@warn@code@too@large` . . . . . 1336, 1339, 1343  
`\MT@warn@err` . . . . . 76, 4106, 4107  
`\MT@warn@ex@too@large` . . . . . 1555, 1558, 1569  
`\MT@warn@maybe@inputenc` . . . . . 2261, 2813, 2821, 2824  
`\MT@warn@nodim` . . . . . 3634, 3647, 3657, 3671  
`\MT@warn@number@too@large` . . . . . 2695, 2805  
`\MT@warn@preset@twidth` . . . . . 1467, 1487, 1680, 1750  
`\MT@warn@rest` . . . . . 2566, 2603, 2810  
`\MT@warn@tracking@DVI` . . . . . 1793, 4665  
`\MT@warn@unknown` . . . . . 2556, 2572, 2582, 2607, 2818  
`\MT@warn@unknown@once` . . . . . 2836, 2837  
`\MT@warning` . 67, 76, 94, 2190, 2194, 2274, 2491,  
3053, 3082, 3170, 3223, 3318, 3341, 3438,  
3567, 3636, 3684, 3698, 3708, 3852, 3873,  
3944, 4106, 4111, 4208, 4285, 4324, 4326, 4327  
`\MT@warning@nl` . . . . . 76, 88, 95, 272, 765,  
769, 805, 817, 903, 1202, 1314, 1350, 1402,  
1442, 1488, 1570, 1998, 2800, 2806, 2811,  
2819, 2868, 3410, 3609, 3955, 3962, 3973,  
4055, 4107, 4112, 4146, 4153, 4189, 4203,  
4375, 4464, 4517, 4536, 4590, 4606, 4671, 4758  
`\MT@while@num` . . . . . 670, 677, 683, 930–932  
`\MT@with@babel@and@T` . . . . .  
. . . 782, 855–857, 892–894, 4746–4750, 4753  
`\MT@with@package@T` . . . 779, 830, 834, 859, 862–  
864, 866, 871, 878–881, 884, 899, 911, 925,  
926, 928, 942, 953, 954, 2836, 2837, 2879, 4589  
`\MT@xadd` . . . . . 104, 570, 2364, 2369, 2374,  
2377, 2940, 2949, 3007, 3107, 3335, 3514, 3787  
`\MT@xaddb` . . . . . 578, 3534, 3862  
`\MT@xdef@n` . . 386, 3255, 3258, 3287, 3292, 3321,  
3437, 3529, 3551, 3717, 3753, 3883, 3970, 3972  
`\MT@xetex@no` . . . . . 240  
`\MT@xspace` . . . . . 2121, 2136  
`\MT@xspace@` . . . . . 2136  
`\MT@xunicode@false` . . . . . 829  
`\MT@xunicode@true` . . . . . 829, 830, 881  
  
`\normalfont` . . . . . 971, 4546, 4550  
`\nullfont` . . . . . 971  
  
`pdftexcmds` (package) . . . . . 37, 233  
`pdftracingfonts` . . . . . 351  
`\pickup@font` . . . . . 2867, 2869, 2878, 2901–2903  
`pifont` (package) . . . . . 98  
`pinyin` (package) . . . . . 56, 232, 233  
`\protrudechars` . . . . . 192, 4276  
`protrusion` (option) . . . . . 6, 11, 123  
`psnfss` (package) . . . . . 78, 195, 235  
`pstricks` (package) . . . . . 9, 132  
`pxfonts` (package) . . . . . 21, 142, 143

- Q** qfonts (package) ..... 21, 142, 229
- R** ragged2e (package) ..... 26, 80, 136, 233, 234  
 reledmac (package) ..... 52, 235  
 relsize (package) ..... 106
- S** selected (option) ..... 8, 14, 32, 68, 89, 124, 228  
 \selectfont ..... 1941, 2970, 2980, 4213, 4764  
 \set@fontsize ..... 3163  
 \SetExpansion ..... 14, 36, 3400  
 \SetExtraKerning ..... 18, 38, 3458  
 \SetExtraSpacing ..... 19, 39, 3443  
 \SetProtrusion ..... 13, 35, 3384  
 \SetTracking ..... 15, 37, 3423  
 \sfcode ..... 4604  
 shapepar (package) ..... 29, 53  
 \shbscode ..... 1647, 1648, 1659, 1662, 1669
- T** \tagcode ..... 1833, 2260, 2266  
 tempora (package) ..... 21, 143, 235  
 tex4ht (package) ..... 54, 231  
 T<sub>E</sub>X Live (distribution) ..... 28, 38, 39, 229  
 \textls ..... 23,  
 52, 53, 919, 944, 949, 1853, 1937, 1953, 2162  
 \textls\* ..... 23  
 \textmicrotypecontext ..... 22, 48, 915, 2974  
 tikz (package) ..... 53, 223, 233, 235
- U** ucs (package) ..... 94, 232  
 ulgothic (package) ..... 21  
 \UnicodeEncodingName ..... 2778
- V** verbose (option) ..... 9, 31, 32, 125, 228, 229, 232
- W** wordcount (package) ..... 51, 232
- X** XCharter (package) ..... 21, 143, 235  
 xeCJK (package) ..... 27, 98, 99, 234, 235  
 X<sub>Y</sub>T<sub>E</sub>X (engine) ..... 1, 4, 6–8, 12,  
 13, 20, 26, 28–30, 34, 38, 40, 50, 52–54, 82,  
 92, 130, 135, 137, 142, 144, 148, 156, 234, 235  
 \XeTeXcharglyph ..... 2593  
 \XeTeXcountglyphs ..... 712  
 \XeTeXfonttype ..... 2544  
 \XeTeXglyph ..... 1285
- Y** yfonts (package) ..... 24
- Z** zefonts (package) ..... 21, 142
- \rightmarginkern ..... 801  
 \rPCODE ..... 1216, 1257, 1258, 1456, 1459, 1838  
 \showhyphens ..... 4544, 4548  
 shrink (option) .. 8, 15, 32, 126, 133, 135, 230, 232  
 slantsc (package) ..... 174  
 soul (package) ..... 4, 30, 55, 56, 81, 231, 233  
 soulutf8 (package) ..... 30, 233  
 \spacefactor ..... 2065  
 \spaceskip ..... 2066, 2069  
 spacing (option) ..... 6, 11, 26, 31, 124, 233, 234  
 \stbscode ..... 1640, 1641, 1658, 1662, 1668  
 step (option) . 8, 31, 32, 126, 133, 135, 228, 230, 232  
 stretch (option) ... 8, 15, 25, 32, 126, 133, 135, 232  
 tikzposter (package) ..... 234  
 tipa (package) ..... 98  
 trace (package) ..... 99, 231  
 \tracingmicrotype ..... 87  
 \tracingmicrotypeinpdf ..... 98  
 \tracingmicrotypeinpdfallfalse ..... 105  
 \tracingmicrotypeinpdfalltrue ..... 105  
 tracking (option) ..... 6, 11, 23, 31, 124  
 txfonts (package) ..... 21, 142, 143  
 unit (option) ..... 7, 32, 229, 231  
 \UseMicrotypeSet ..... 12, 33, 60, 73, 3047, 3234

## 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 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

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

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 modified versions of your work that conform with whatever technical specifications you wish while maintaining the

availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document '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

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating
  - (a) a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
  - (b) If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
    - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
    - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
    - (c) No information in the Derived Work implies that

- any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
- (d) You distribute at least one of the following with the Derived Work:
- i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
  - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
- (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.
11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

## No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any au-

thor named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

## Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other

significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
  - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.

- (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L<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
% http://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.